**Team Members:**

**Sri Vishnu. M :** [**srivishnu1103@gmail.com**](mailto:srivishnu1103@gmail.com)  **(Team Leader)**

**Sathis Kumar. G :** [**g.sathiskumar22@gmail.com**](mailto:g.sathiskumar22@gmail.com)

**Sanjay Kumar. R :** [**rsanjaykumar1319@gmail.com**](mailto:rsanjaykumar1319@gmail.com)

**Mohammed Noorulla :noorullasnoorul123@gmail.com**

Fake News Detection using NLP

In this machine learning project, we build a classifier that detects whether the news is fake or not.

This is a binary classification problem. We preprocess the text data from our dataset using TF-IDF Vectorizer. We apply the Multinomial Naive Bayes algorithm to the preprocessed text and train and evaluate our model on the dataset.

In this project, we have used various natural language processing techniques and machine learning algorithms to classify fake news articles using sci-kit libraries from python.

Dataset used:

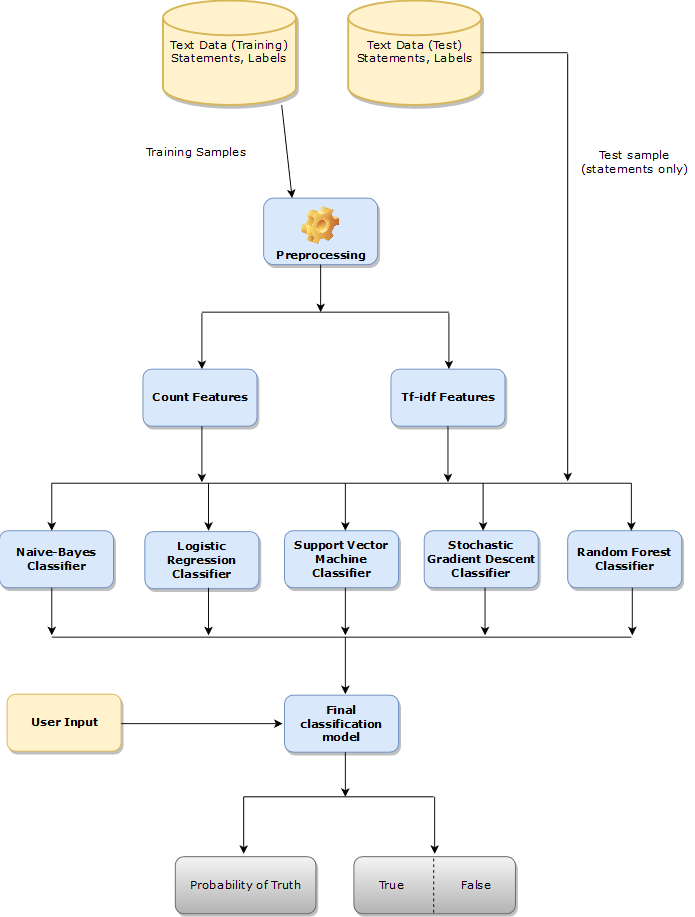
Kaggle dataset: true dataset and fake dataset

True dataset: The articles in the dataset are true.

Fake dataset: The articles in the dataset are fake.

The following flowchart represents the design thinking of this project.

Below is the Process Flow of the project:



The Steps involved are explained as follows,

1. Data Collection:

Gather a diverse dataset of news articles, including both real and fake news, to train and test your model.

2. Text Preprocessing:

Text preprocessing involves transforming text into a clean and consistent format that can then be fed into a model for further analysis and learning.

3. Feature Extraction:

Convert text into numerical features. Common methods include TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings like Word2Vec or Glove.

4. Model Selection:

Choose an appropriate machine learning or deep learning model. Common choices include Logistic Regression, Random Forest, Support Vector Machines, Naive baye’s classifier, Stochastic gradient descent and neural networks like LSTM or BERT.

5. Model Training:

Train your selected model on the preprocessed data, using labelled examples of real and fake news.

6. Model Evaluation:

Use evaluation metrics like accuracy, precision, recall, F1-score, and ROC-AUC to assess the model's performance on a separate test dataset.

7. Fine-Tuning:

Experiment with hyperparameters and model architectures to optimize performance.

8. Cross-Validation:

Implement cross-validation techniques like k-fold cross-validation to ensure your model generalizes well.

9. Post-processing:

Apply additional techniques like thresholding or ensemble methods to refine predictions.

10. Real-time Monitoring:

If deploying for real-time monitoring, set up a system to continuously analyze incoming news articles and classify them as real or fake.

11. User Interface:

Develop a user-friendly interface or application for users to check the authenticity of news articles.

12. Updates and Maintenance:

Regularly update your model with new data to adapt to involving fake news tactics.

13. Ethical Considerations:

Consider the ethical implications of your fake news detection system, such as biases in the data and model, and take steps to mitigate them.

Loading and preprocessing the dataset are essential steps. Here's a high-level overview of the process:

1. Loading the Dataset:

Obtain a labeled dataset that contains news articles or text data with labels indicating whether they are fake or real news.

Common sources for such datasets include Kaggle, research repositories, or custom data collection efforts.

2. Data Cleaning:

Remove any irrelevant or extraneous information from the text, such as HTML tags, special characters, or numbers.Tokenize the text into words or subwords, depending on your NLP model.

3. Text Preprocessing:

Convert text to lowercase to ensure uniformity.

Remove stop words (common words like "the," "and," "in") to reduce noise.Perform stemming or lemmatization to reduce words to their root form.

4. Data Splitting:

Split the dataset into training, validation, and test sets. A common split might be 70% for training, 15% for validation, and 15% for testing.

5. Padding:

If you're using neural networks, pad or truncate the sequences to a fixed length to ensure consistent input sizes.

6. Data Loading:

Use data loaders or generators to efficiently load and feed the data to your NLP model in batches. Libraries like PyTorch and TensorFlow provide tools for this.

7. Label Encoding:

Encode the labels (fake or real) as numerical values, typically 0 and 1.

8. Balancing the Dataset (Optional):

Check for class imbalance and consider techniques like oversampling or under sampling to balance the dataset.

9. Data Augmentation (Optional):

For text data, data augmentation techniques are less common than in image processing, but you can consider methods like synonym replacement or adding noise to text to increase dataset variety.

Once you've completed these preprocessing steps, you can use the preprocessed data to train and evaluate your fake news detection NLP model, which could be based on various algorithms such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), or transformer-based models like BERT.

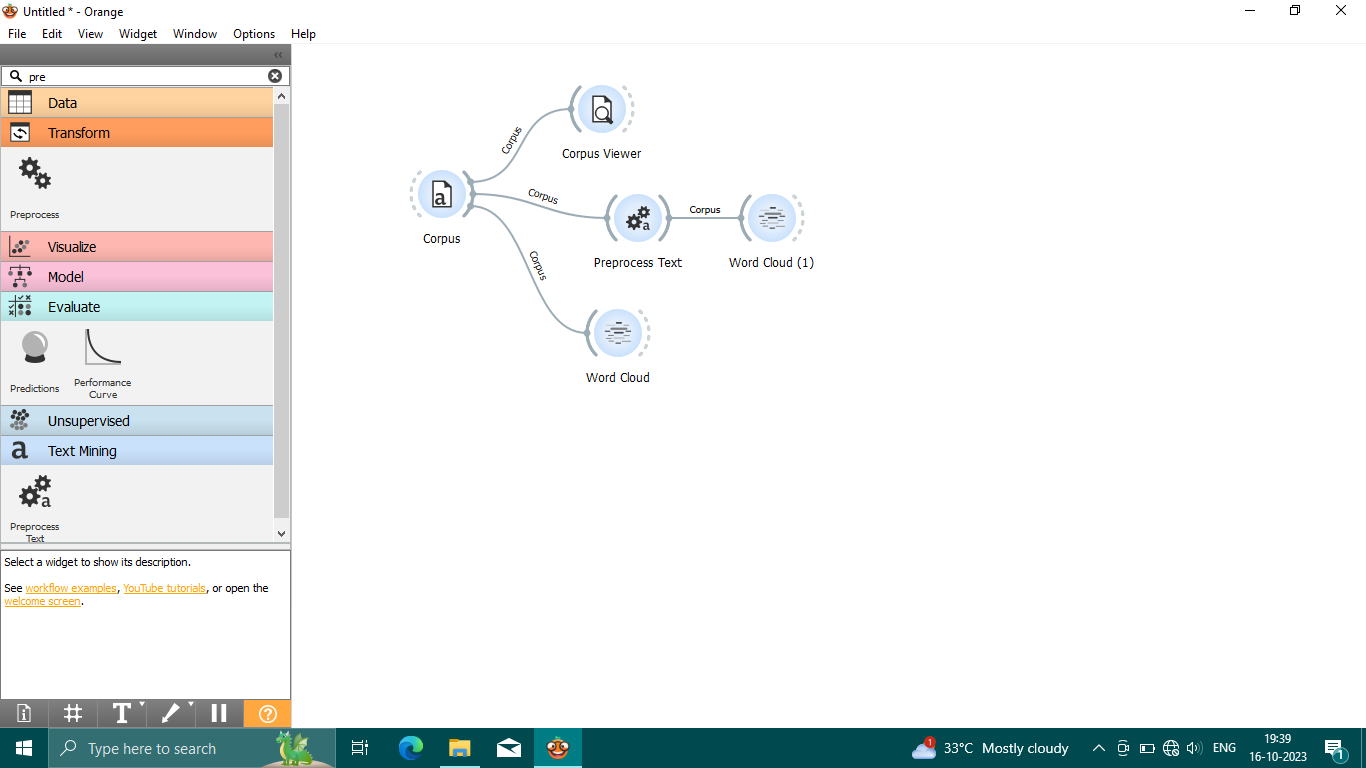
Remember that the specific preprocessing steps may vary depending on the characteristics of your dataset and the NLP model you're using.

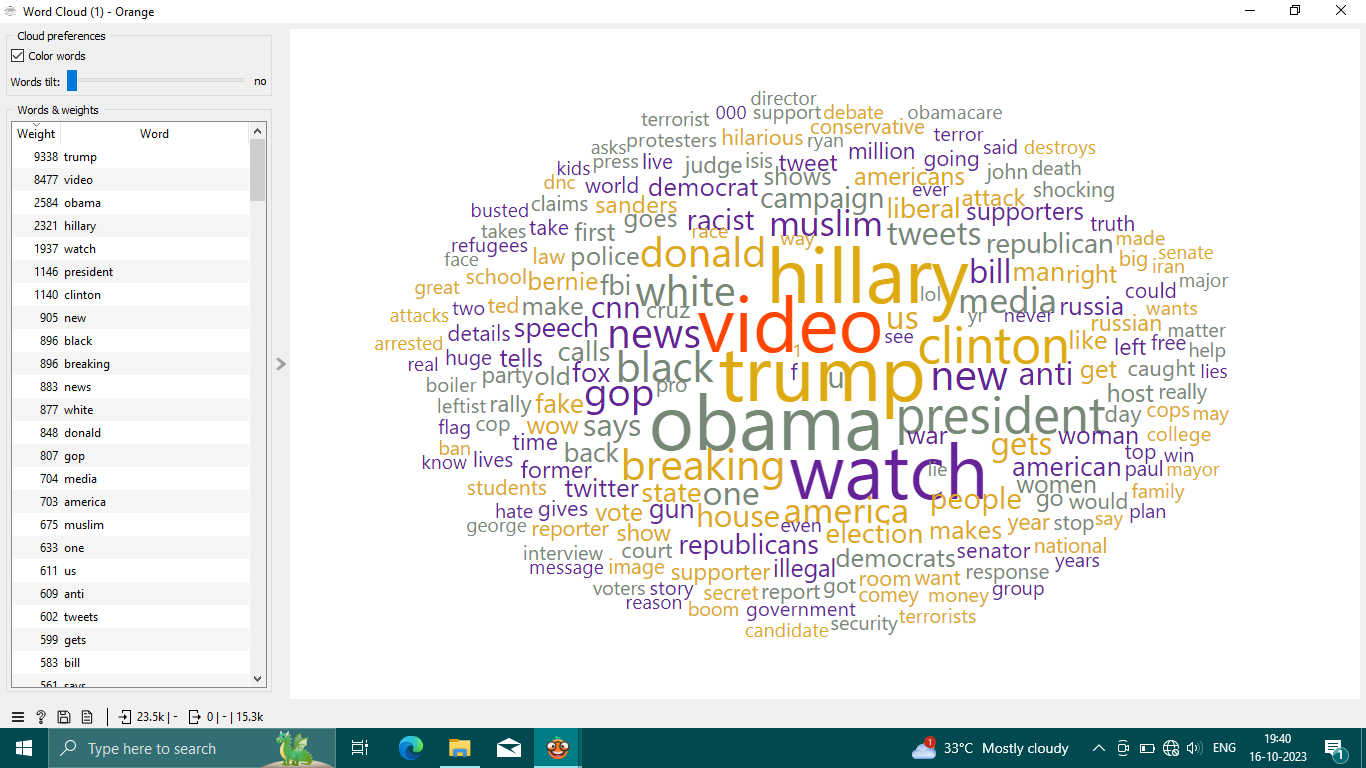
For loading and preprocessing here we are using orange software or tool.

The following images represents the loading and preprocessing of the kaggle data set using orange tool.

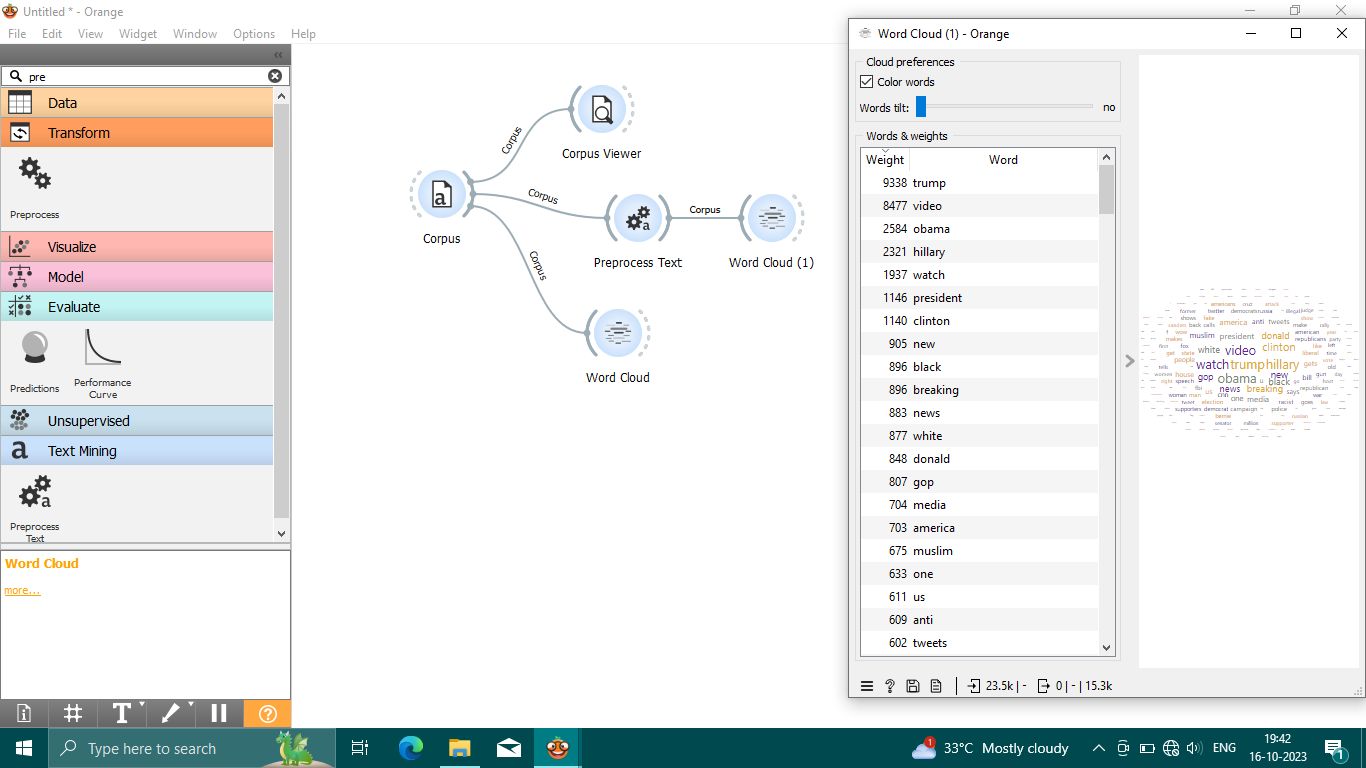
Loading and Preprocessing using orange software:

For fake datasets.

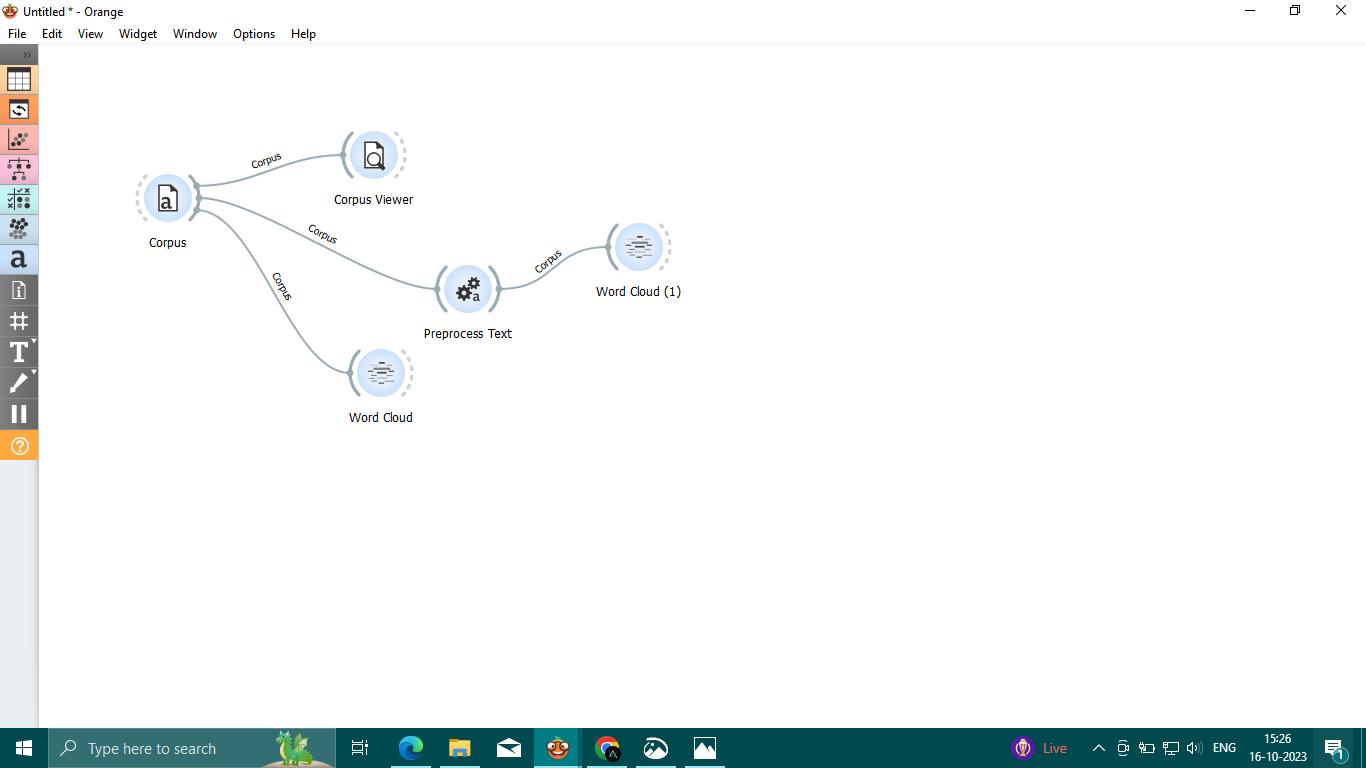


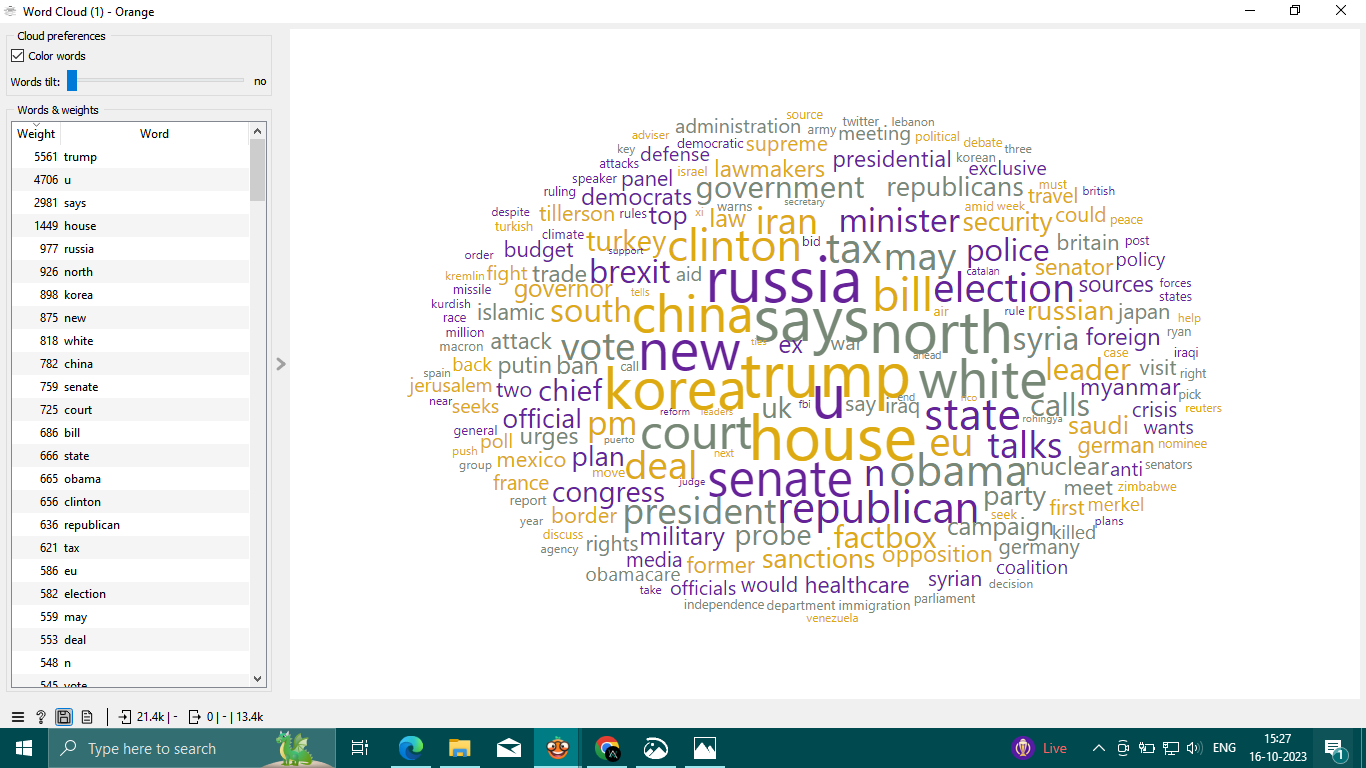


Word cloud

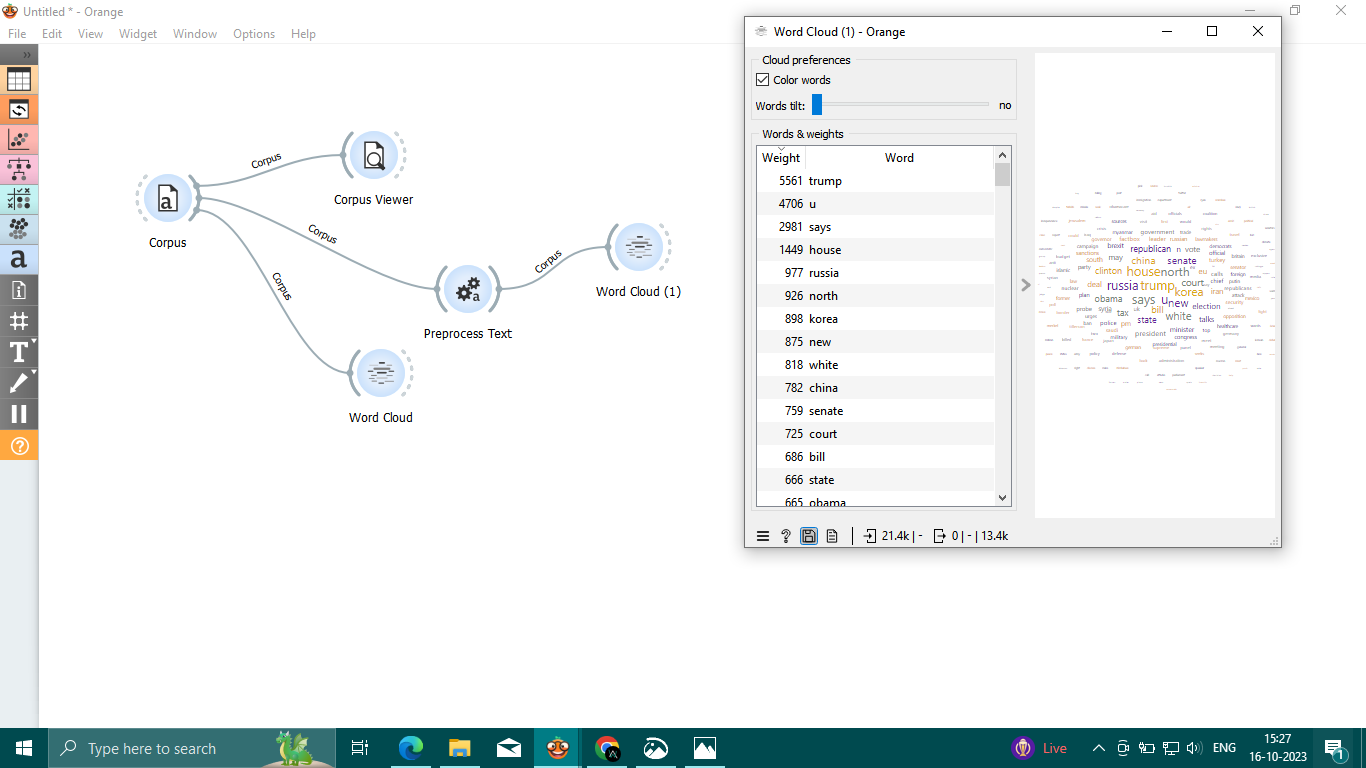


For true datasets.





Word cloud



It also involves several key steps:

•Feature engineering

•Model training

•Evaluation.

Here's a high-level overview of each step:

Feature Engineering:

Text Preprocessing:

Clean and preprocess the text data, including tasks like tokenization, lowercasing, and removing stop words, punctuation, and HTML tags.

Text Representation:

Convert text data into numerical form. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings like Word2Vec or GloVe.

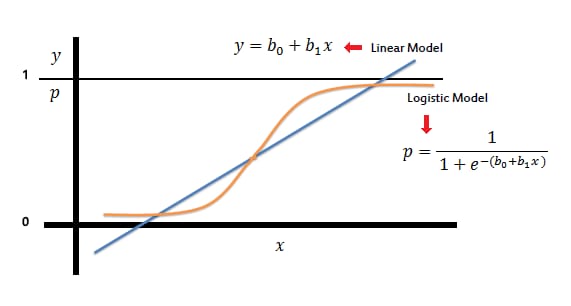
Feature Selection:

Choose relevant features. This can involve using techniques like mutual information or feature importance scores.

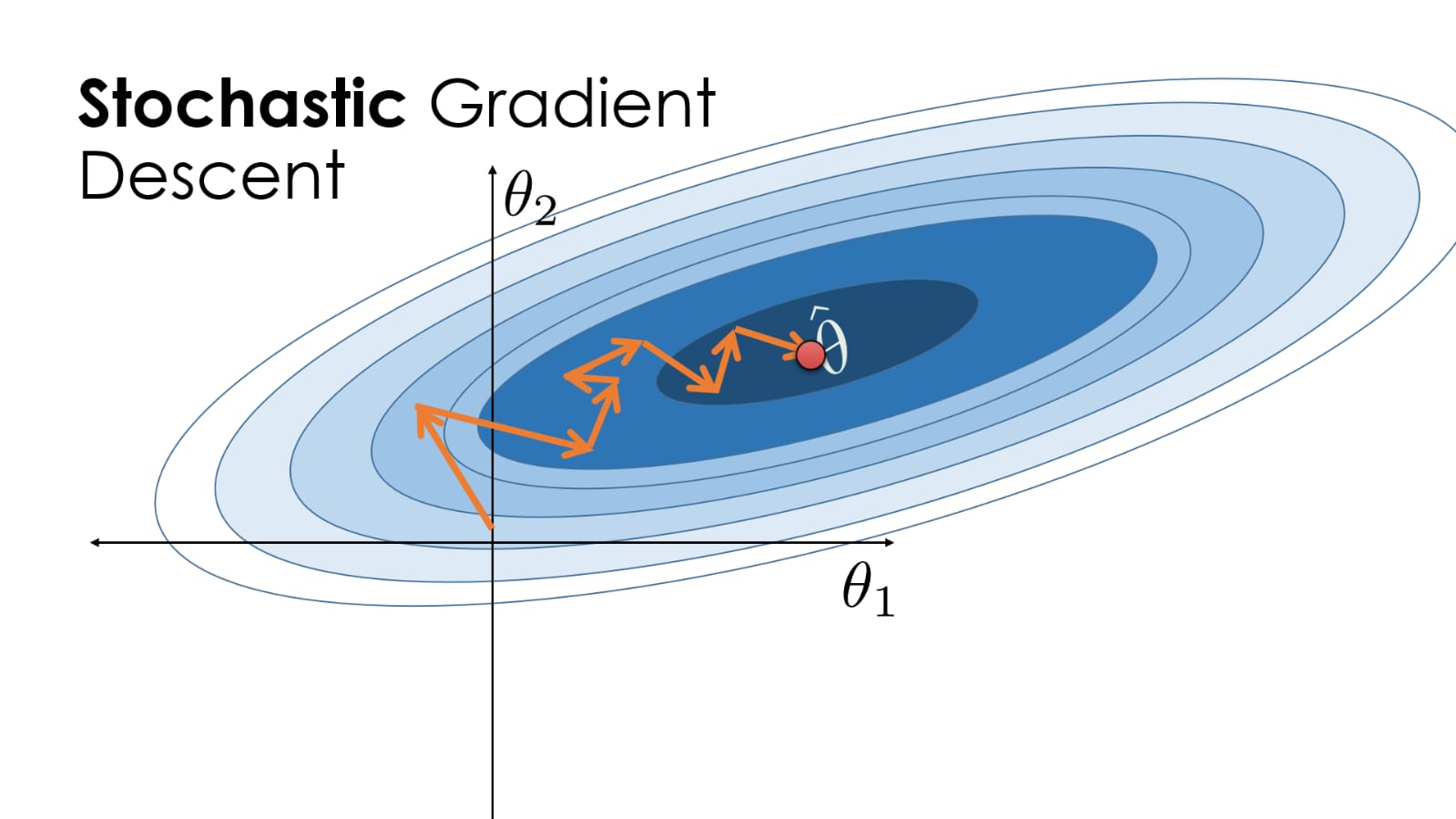
Model Training:

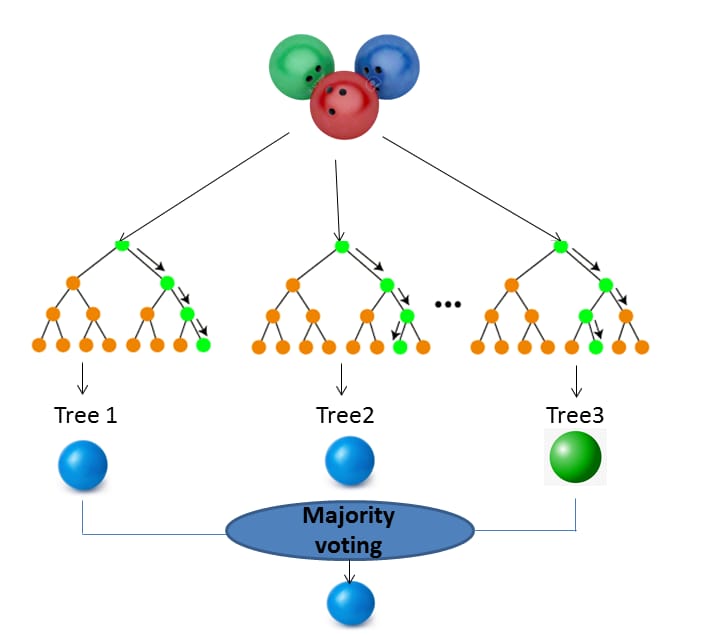
Selecting Models:

Choose appropriate machine learning or deep learning models for your task. Common choices include logistic regression model, stochastic gradient descent, random forest, support vector machine, naïve bayes.

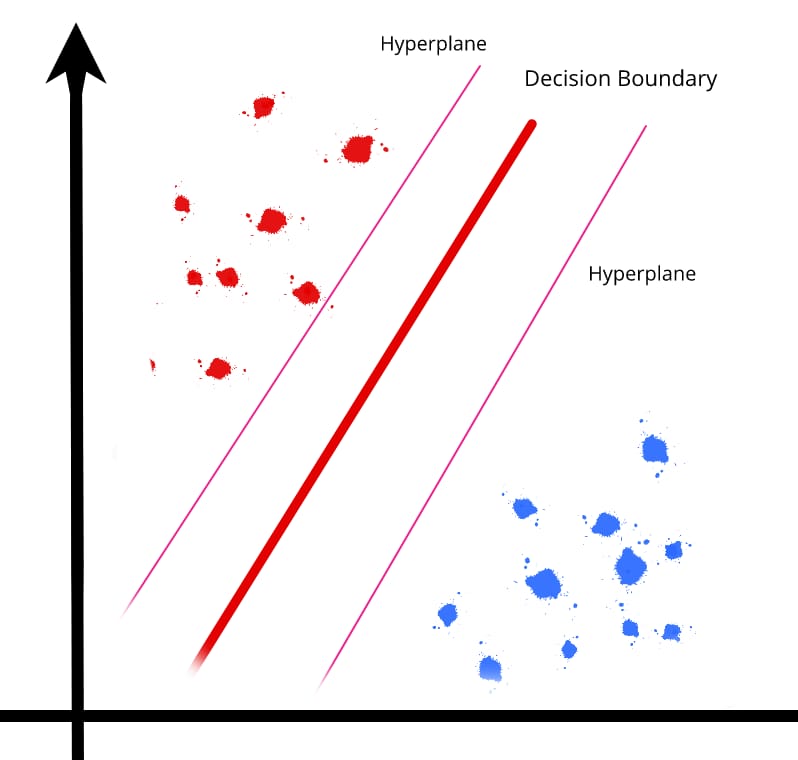


Logistic regression model

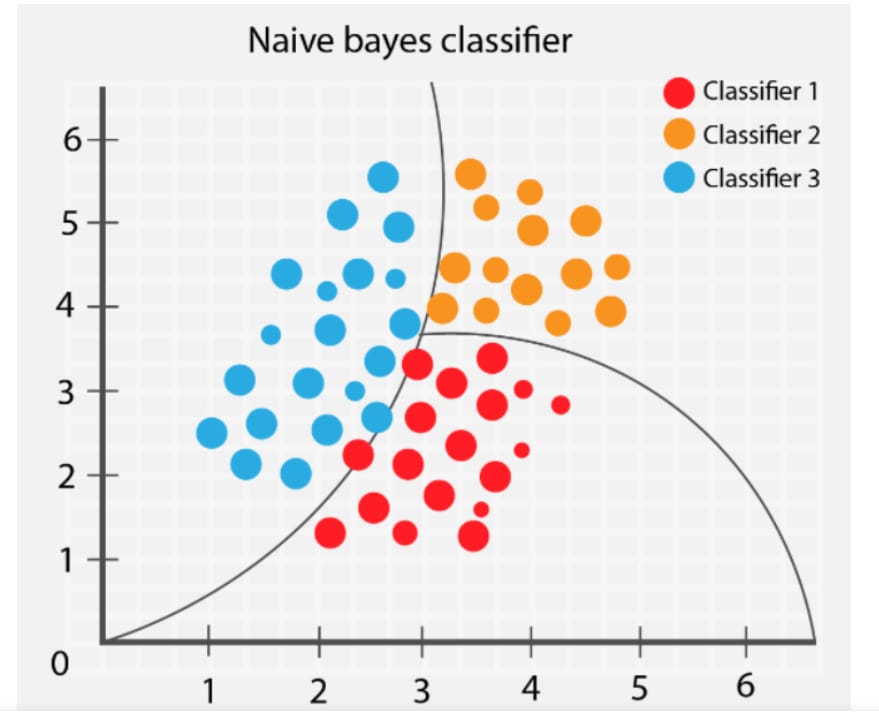
**

**

Random forest

**

Support vector machine

**

Naïve bayes classifier

Data Splitting:

Divide your dataset into training, validation, and test sets to train and evaluate the model.

Model Training:

Train your selected model on the training data using the chosen features and representations.

Hyperparameter Tuning:

Optimize hyperparameters to improve the model's performance using techniques like grid search or random search.

Evaluation:

Metrics:

Choose appropriate evaluation metrics. Common metrics for binary classification problems like fake news detection include accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).

Confusion Matrix:

Analyze the model's performance using a confusion matrix to understand true positives, true negatives, false positives, and false negatives.

Cross-Validation:

Perform k-fold cross-validation to assess the model's stability and generalization.

Bias and Fairness:

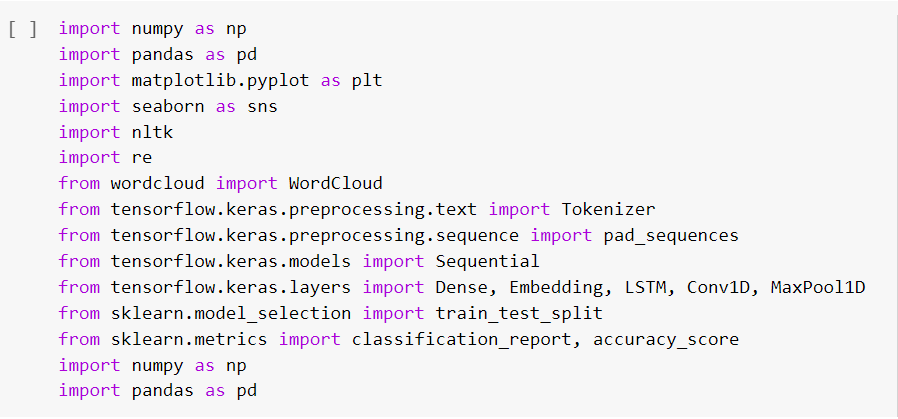
Check for bias and fairness issues in your model's predictions to ensure that it's not exhibiting discriminatory behavior.

Ensemble Methods:

Consider using ensemble techniques like bagging or boosting to improve model performance further.

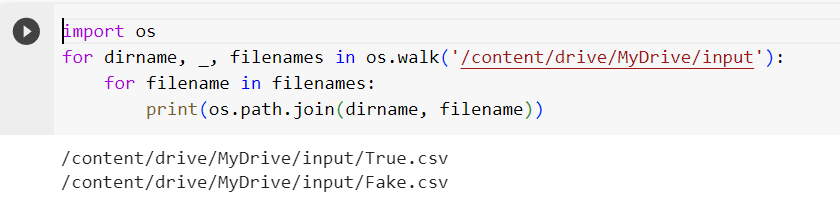
In the first step, we will only remove the unnecessary data points from the dataset which does not helps in improving the model performance.

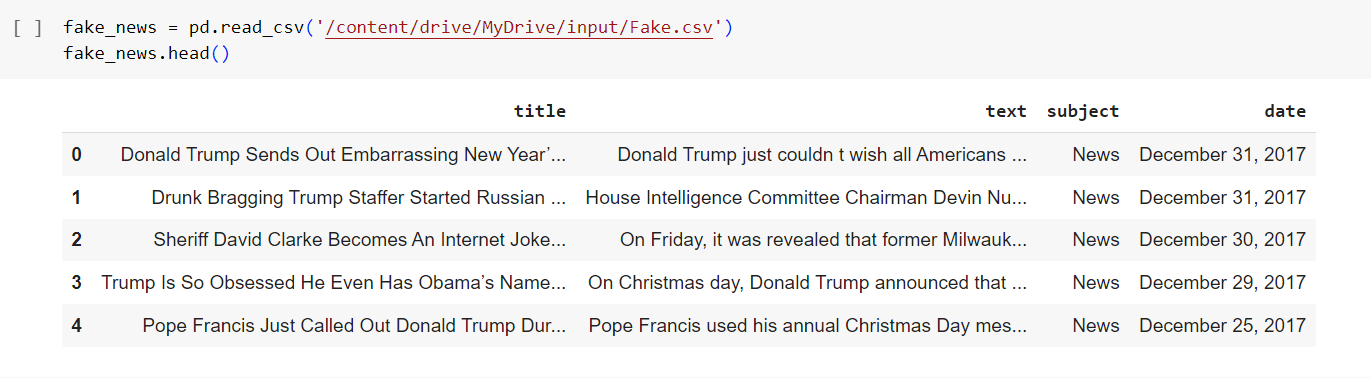
Initially we import the necessary packages for our data cleaning process and also in the future purposes,

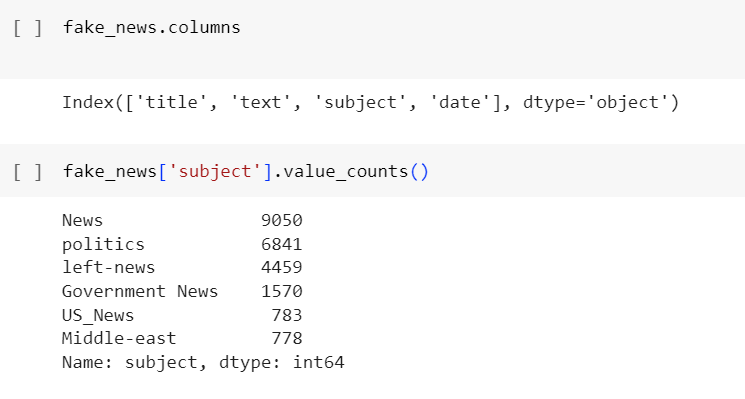


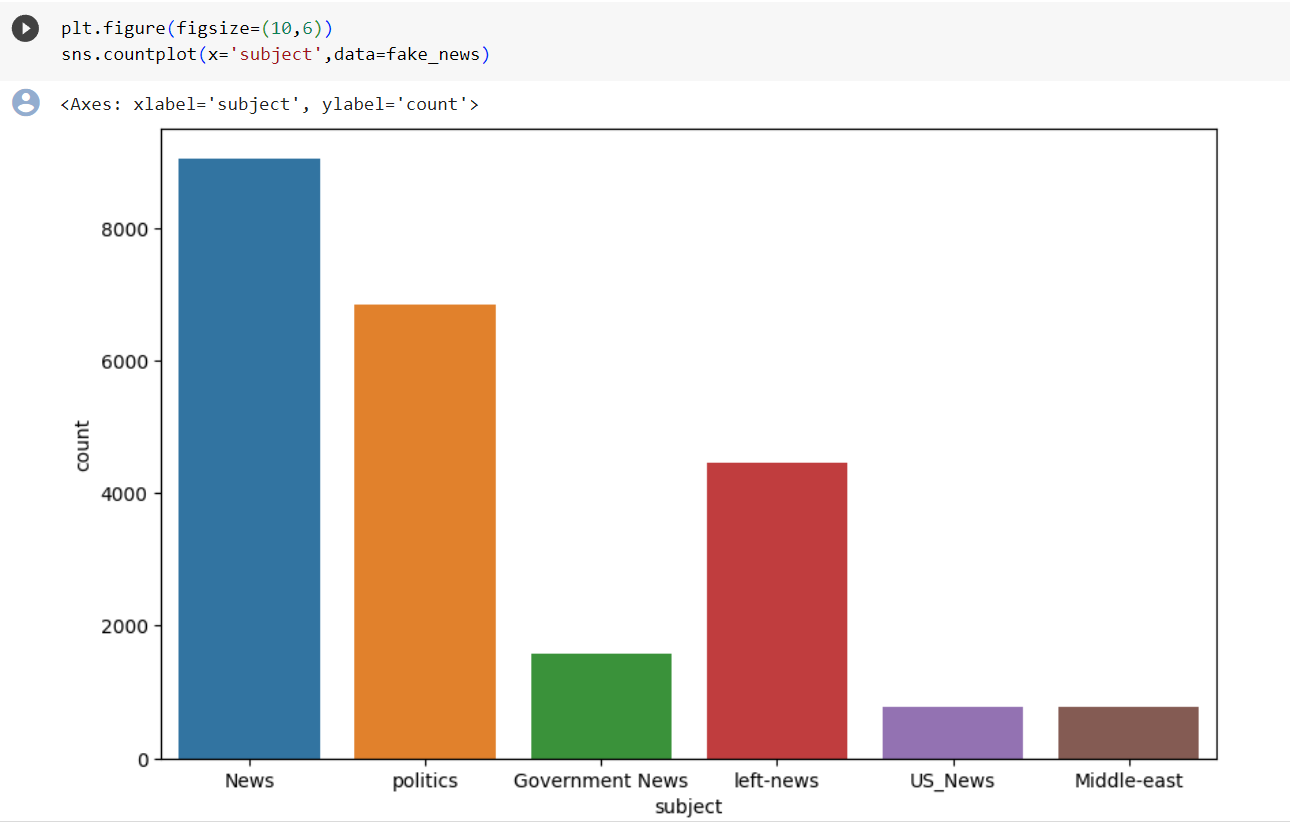
we use these packages in various stages of our cleaning process and also in the future in which we need to build models.

Here, we read the .csvfiles of true and fake news and then explore the count values of their subjects











Let’s create a list of news lists in real\_news.csv with unknown publishers by using the following code snippets

