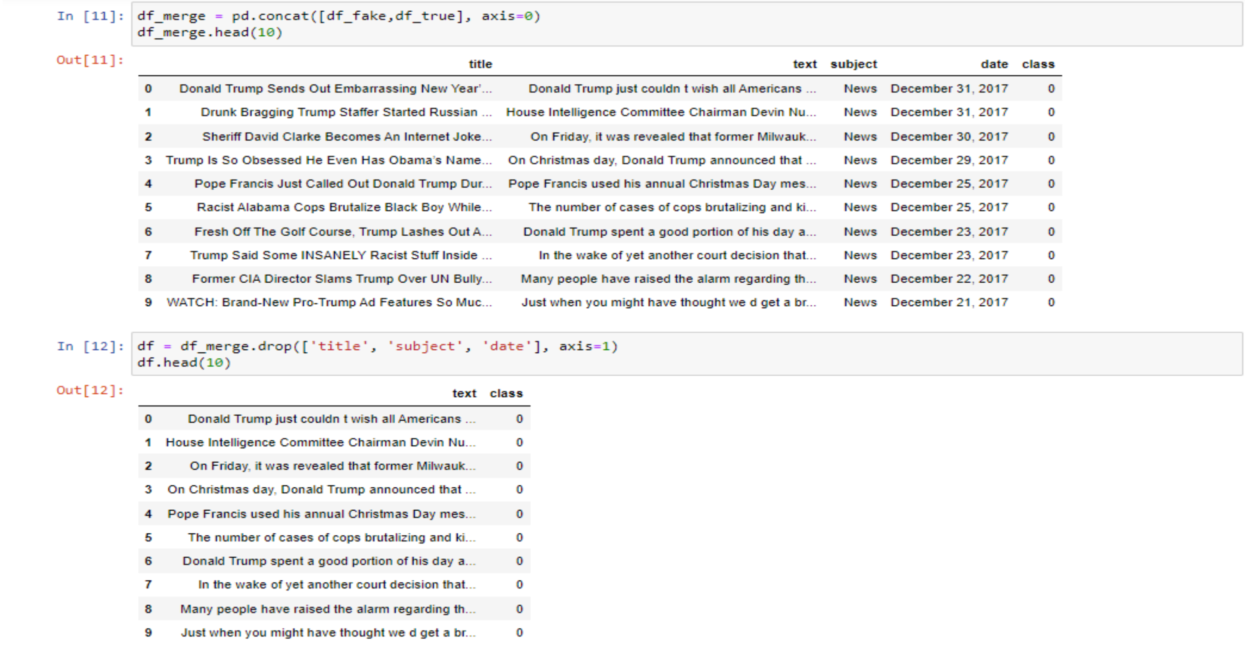
Haseeb Chaudhury
CSC44800
11/5/23

Exploring supervised learning methods using fake news dataset

Abstract

News is one of the most prominent sources of information in our world currently. Many people listen to different types of news and can come up with their conclusions and such. However, not all the news people watch is considered real. Fake news existed before from social media and other platforms when it was introduced. The purpose of creating such fake news was to deliberately spread false information(misinformation), create hoaxes and spread certain types of propaganda. This is usually done through social media sites such as Facebook, Twitter, etc, and is often done to further certain ideas and is often achieved with political agendas. Fake news has influenced the perception of an individual and this has increasingly become a major problem in contemporary society. It's very difficult to determine whether a text is factual without additional context and human judgment. The purpose of this paper is to utilize the fake and real news detection dataset and train and test my model using various supervised methods and will compare the accuracy of all of the models. We will use various techniques such as TF-IDF vectorizer, the passive-aggressive classifier, logistic regression, random forest, and many more algorithms. I will be using various models and algorithms to come up with predictions on determining whether certain news is fake or not. We will explore each supervised learning method in detail and explain how it relates to my dataset.

My Dataset



In [11]:
```
df_merge = pd.concat([df_fake,df_true], axis=0)
df_merge.head(10)
```

Out[11]:

	title	text	subject	date	class
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0
5	Racist Alabama Cops Brutalize Black Boy While...	The number of cases of cops brutalizing and ki...	News	December 25, 2017	0
6	Fresh Off The Golf Course, Trump Lashes Out A...	Donald Trump spent a good portion of his day a...	News	December 23, 2017	0
7	Trump Said Some INSANELY Racist Stuff Inside ...	In the wake of yet another court decision that...	News	December 23, 2017	0
8	Former CIA Director Slams Trump Over UN Bully...	Many people have raised the alarm regarding th...	News	December 22, 2017	0
9	WATCH: Brand-New Pro-Trump Ad Features So Muc...	Just when you might have thought we d get a br...	News	December 21, 2017	0

In [12]:
```
df = df_merge.drop(['title', 'subject', 'date'], axis=1)
df.head(10)
```

Out[12]:

	text	class
0	Donald Trump just couldn t wish all Americans ...	0
1	House Intelligence Committee Chairman Devin Nu...	0
2	On Friday, it was revealed that former Milwauk...	0
3	On Christmas day, Donald Trump announced that ...	0
4	Pope Francis used his annual Christmas Day mes...	0
5	The number of cases of cops brutalizing and ki...	0
6	Donald Trump spent a good portion of his day a...	0
7	In the wake of yet another court decision that...	0
8	Many people have raised the alarm regarding th...	0
9	Just when you might have thought we d get a br...	0

The dataset that I will be using is the fake and real news dataset which I got from Kaggle.

Testing and training

began testing and training my dataset and I will be testing 25% of it. Here I used a TF-IDF vectorizer from sci-kit to learn to vectorize my text variables by fitting training data and then transforming my testing data. TF-IDF is defined as term frequency and inverse document frequency. Term frequency is the number of times a word appears in a document. A higher value means a term appears more often than others which means that the document is a good match when the term is part of the search terms. IDF stands for inverse document frequency which are words that occur many times in a document, but also occur many times in many others. Therefore, IDF is a measure of how significant a term is in the entire document. So The

TfidfVectorizer converts a collection of raw documents into a matrix of TF-IDF features. The formula for TF is TF(x)= (Number of times word x appears in a document)/ (Total Number of words in a document) and for IDF is IDF(x) = log_e(Total number of documents)/(Number of documents with the word x in it). TFIDF is Commonly used in data mining and data cleaning where search engines use it frequently to rate and rank documents as well as being used to stop a variety of stop words. Basically, this gives us information on how often a term appears in a document and IDF gives us information about the relative rarity of a term in the collection of documents. TF-IDF can be used for determining how important a term is in a document. This can be used to help summarize articles more efficiently for a document. The biggest advantages of TF-IDF are that it's easy to use and it's cheap.

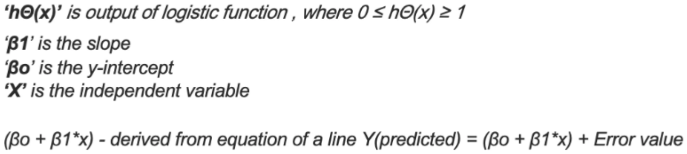
PassiveAggresiveClassifier

The class of online learning algorithms in machine learning includes passive-aggressive classifiers. It operates by acting passively in response to accurate classifications and aggressively in response to incorrect classifications. A machine learning model is built and put into use in online learning so that it can keep learning as new data sets are added. We can therefore conclude that systems that receive data in a continuous stream benefit most from an algorithm like the passive-aggressive classifier. Its goal is to make updates that fix the loss while barely changing the weight vector's norm. They are generally used in large-scale learning. For example, we see this being used to detect fake news on social media websites like Twitter where new data is constantly being added every second. I imported this algorithm from sci-kit learn and then assigned the variable with a random_state of 0, then fitted it using my training and testing data. Note that all of the algorithms that I will discuss comes from sci-kit-learn and will be defined the same way as shown below by importing them, creating a variable with random_state 0, and fitting the model using training and testing data.

Logistic Regression

Logistic regression is a type of supervised learning. This is used to predict the probability of a binary event that occurs. In this project, we are trying to determine whether certain news is considered fake or real, which are 2 different outcomes that are defined as binary classification since there are 2 possible outcomes. Logistic regression is used to solve a variety of binary classification problems.

$$h\Theta(x) \;=\; 1\,/1 \;+\; e - (\beta o \;+\; \beta 1X)$$



'hΘ(x)' *is output of logistic function , where 0 ≤ hΘ(x) ≥ 1*
'β1' *is the slope*
'βo' *is the y-intercept*
'X' *is the independent variable*

*(βo + β1*x) - derived from equation of a line Y(predicted) = (βo + β1*x) + Error value*

According to mathematics, probability always falls between 0 (does not happen) and 1(happens). The likelihood of testing false and not testing fake in our news detection example will add up to 1. In logistic regression, the probability is calculated using the logistic function or the sigmoid function. When there are only two possible outcomes, as in our example of whether the news is regarded to be fake or real, we use binary logistic regression. Multinomial logistic regression is used when there are more than two possible outcomes. For instance, in our example, we may expand to include the possibility of outcomes depending on whether specific news categories are based on dates or subjects. In classification situations when the output or dependent variable is binary or categorical, logistic regression is applied. When using logistic regression, we need to understand the different types of independent variables and the training data available.

Naive Bayes

The Bayes' Theorem is a machine learning algorithm that is used by all Naive Bayes classifiers to categorize data points. In order to forecast the outcomes of brand-new data points, Naive Bayes classifiers employ the probabilities that specific events will occur, assuming that other events will occur as expected. This is what distinguishes this formula from other machine learning classifying formulas. Some advantages are that it is very straightforward to construct/use, simple to train, and ignores unnecessary features. The disadvantage is that it assumes that the data point features are independent and that large data sets perform better than smaller data sets.

$$P(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

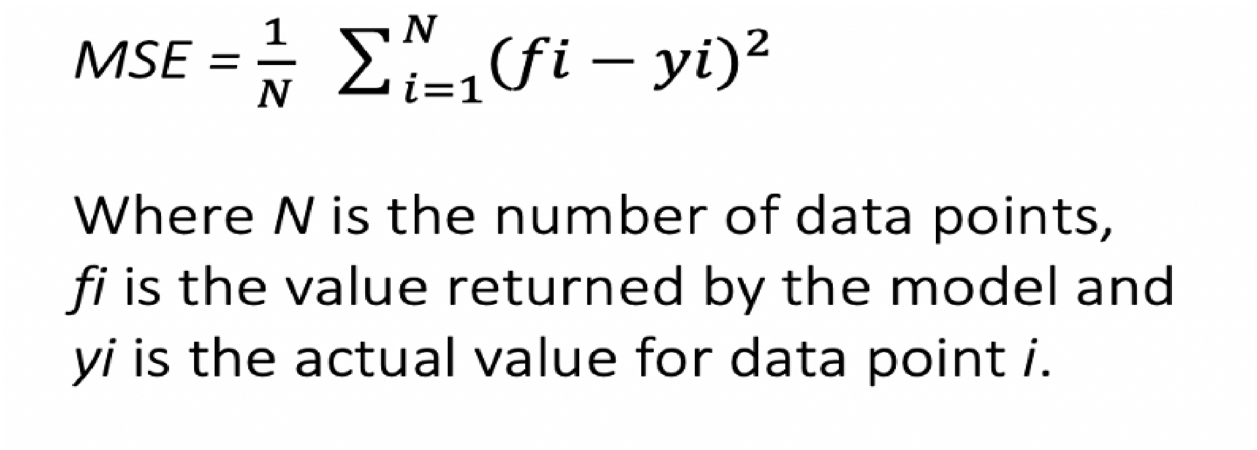
Since the classifiers merely apply the formula to sets of data, the Naive Bayes classifiers are wholly dependent on Bayes' Theorem. This theorem includes a formula for calculating the likelihood that various occurrences will occur. The simplest formulation of it has just two events which are event A and event B and then the Bayes formula is applied. In conclusion, Naive Bayes is a class of techniques that can be used to classify big data sets using probability.

Decision Trees

A decision tree is a supervised machine-learning tool that may be used to classify or forecast data based on how queries from the past have been answered. The building is the process of creating a decision tree, during which you choose the circumstances and qualities that will result in the tree. The tree is then trimmed to remove unnecessary branches that can interfere with accuracy. Pruning entails identifying outliers, or data points that deviate greatly from the norm, which could cause computations to be incorrect by overly emphasizing unusual events in the data. They can be used to address both classification and regression issues. The objective is to learn straightforward decision rules derived from the data features in order to build a model that predicts the value of a target variable. Its ability to operate with numerical or categorical data and variables, as well as how simple the algorithm is to comprehend, are some of its benefits. The drawback is that it's not the best option for big datasets and it can unequally weight or value attributes. The binary nature of the decisions at nodes limits the complexity that the tree can manage. When dealing with uncertainty and several linked outcomes, trees can get exceedingly complex. When attributes can be compared to predetermined criteria to determine the final category, decision trees are helpful for categorizing the results. Decision trees represent the potential outcomes of a set of connected decisions.

RandomForest

Using supervised learning, the random forest algorithm generates results. This "learns" how to categorize unlabeled data using labeled data. Engineers frequently utilize the Random Forest Algorithm because it may be used to address classification and regression difficulties. Its application in classification and regression issues makes it a versatile model, which is one of its benefits as well as preventing data from being overfitting and having rapid test data training. Some of the cons are that once a model is created, it takes a while for predictions to be made, and you must watch out for anomalies and data gaps. You use the mean squared error (MSE) to determine how your data branches from each node while utilizing the Random Forest Algorithm to solve regression problems.



$$MSE = \frac{1}{N} \sum_{i=1}^{N} (fi - yi)^2$$

Where *N* is the number of data points, *fi* is the value returned by the model and *yi* is the actual value for data point *i*.

To determine which branch is the best choice for your forest, this formula estimates the distance between each node and the expected actual value. In this case, fi is the value the decision tree returned, and Yi is the value of the data point you are testing at a particular node.

Before you can completely comprehend the random forest method, you must first understand a single decision tree. To reach a conclusion, you must comprehend the distinctions between a node, branch, and leaf as well as how the various formulas are used. The random forest algorithm is very useful when dealing with different datasets. Furthermore, it is simple to use, faster to train, and can find an accurate representation of the decision trees it is using.

KNN Neighbors

Using the points that are most similar to them, the KNN model categorizes data points. It "informed guesses" what an unclassified point should be classed as using test data. Its simplicity of usage, rapid computation time, and not assuming anything about the data are the pros. The cons is that accuracy is reliant on data quality and must determine the best k value (number of nearest neighbors), and it is unable to accurately classify data points that lie on a line where they can be categorized in one of two ways. KNN is a non-parametric algorithm that serves as an illustration of lazy learning. It is non-parametric if it contains no presumptions. Instead of assuming that the structure of the model is normal, the model is totally constructed from the data provided to it. The algorithm does not generalize anything because it uses lazy learning. This indicates that adopting this strategy requires little training. As a result, when employing KNN, all of the training data is also used in testing. KNN functions as it does because it is based on fundamental mathematical theories, just like practically everything else. The initial step in using KNN is to convert data points into feature vectors, or their numerical value. The method then calculates the separation between these points' mathematical values. The Euclidean distance, as displayed here, is the most typical method for calculating this distance.

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

To determine the separation between each data point and the test data, KNN applies this formula. It then calculates the likelihood that these points correspond to the test data, and it categorizes the data according to which points have the highest probabilities in common.

SGD Loss functions

Stochastic Gradient Descent (SGD) is a simple approach to fitting linear classifiers and regressors under different types of loss functions. SGD is used in large-scale learning and has been successfully utilized in large-scale machine learning problems dealing with text classification and natural language processing. In the SGD, it can scale up to 100000 training examples and features according to its classifiers. Some of the pros of using Stochastic Gradient Descent are the efficiency and its very simple use. The cons of Stochastic Gradient Descent are that it can require a number of hyperparameters including iterations, and regularization, and it can be sensitive toward feature scaling. I used the SGD Square, log_loss, hinge, and modified Huber imported from the sci-kit-learn library.

SVM(Support Vector Machines)

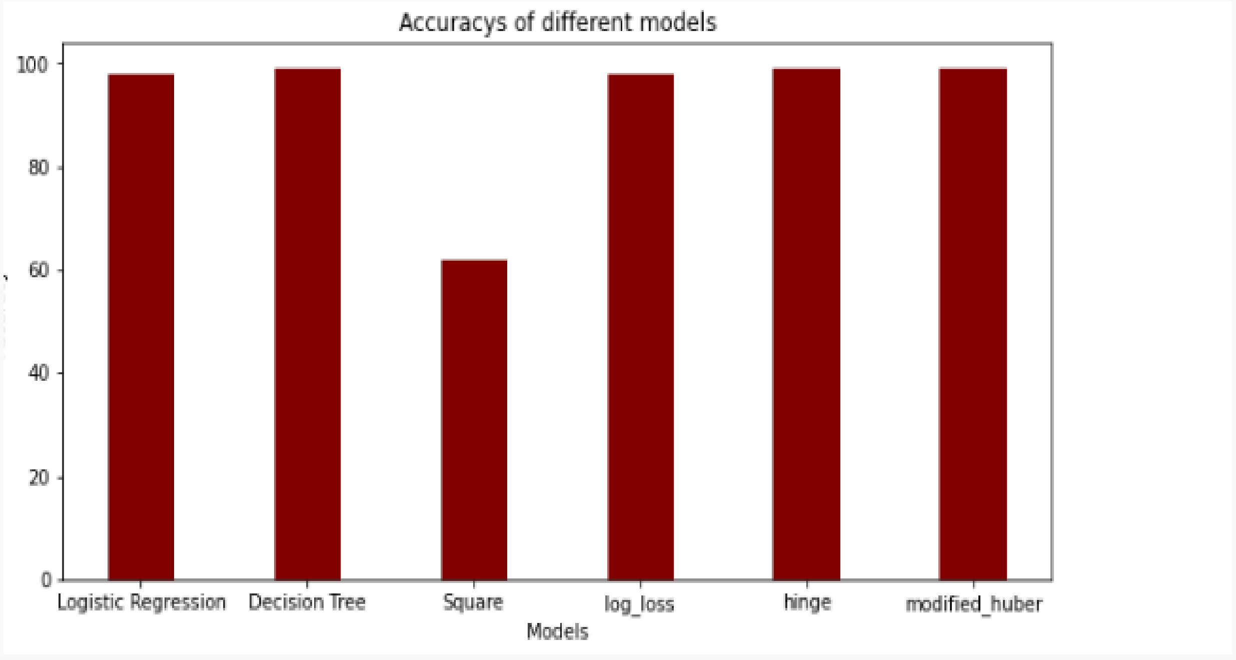
Support vector machines (SVMs) are a group of supervised learning techniques for classifying data, performing regression analysis, and identifying outliers. Support vector machines' benefits include efficiency in high-dimensional environments. It is also memory efficient because it only uses a portion of the training points (known as support vectors) in the decision function. Different Kernel functions can be given for the decision function, making it versatile. There are common kernels available, but you can also define your own kernels. Support vector machines have a variety of drawbacks, including the need to avoid over-fitting when selecting Kernel functions and regularization terms if the number of features exceeds the number of samples. SVMs aren't primarily responsible for probability estimates, instead, they are calculated by using a five-fold cross-validation. From the SVM A hyperplane is used to separate data into different classes. The vectors are then utilized to make sure that the hyper-plane has the largest margin. The benefit of this is that it can draw in high accuracy for certain classification problems and this algorithm can be applied to many different cases such as the one in this situation being the detection of fake news The math is mainly incorporated by the SCIkit-learn library.

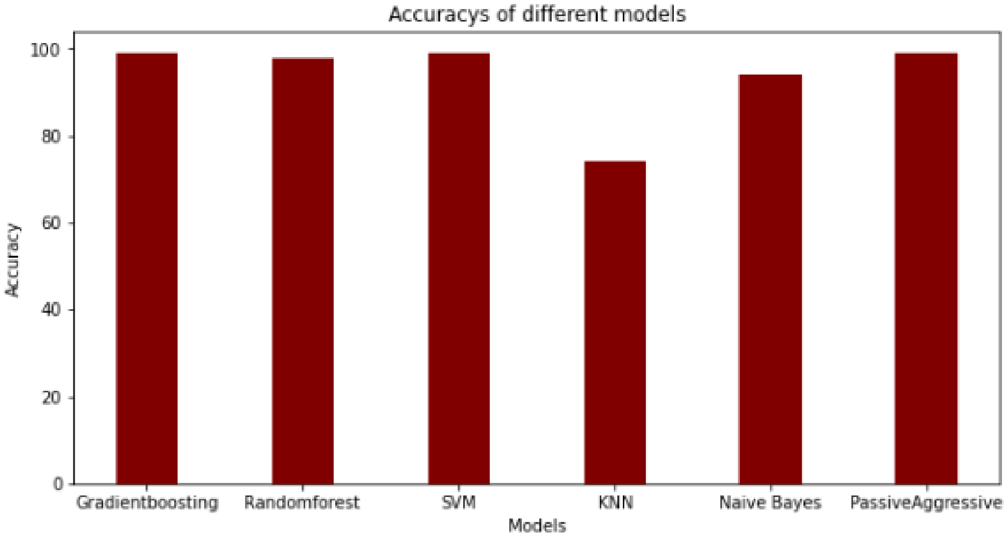
Gradient Boosting for classification.

Gradient boosting is one of the most powerful techniques for building predictive models. It involves 3 elements which are a loss function to be optimized, a weak learner to make predictions, and an additive model to add weak learners to minimize the loss function. The loss function used depends on the type of problem being solved. A benefit of the gradient boosting framework is that a new boosting algorithm does not have to be derived for each loss function that may want to be used, instead, it is a generic enough framework that any differentiable loss function can be used. Decision trees are used as the weak learner in gradient boosting. When adding trees, a gradient descent approach is utilized to reduce loss. Gradient descent is typically used to reduce a set of parameters, such as the weights in a neural network or the coefficients in a regression equation. The weights are changed to reduce inaccuracy after calculating loss or error. Weak learner sub-models, or more particularly decision trees, are used in place of parameters. We must incorporate a tree into the model to lower the loss before we can begin the gradient descent technique (i.e. follow the gradient). We achieve this by parameterizing the tree, modifying its parameters, and moving in the right direction by (decreasing residual loss. Being a greedy method, gradient boosting can quickly overfit a training dataset. At each level, n classes_ regression trees are fitted to the negative gradient of the loss function, such as a binary or multiclass log loss. One regression tree is constructed specifically for the binary classification case.

Conclusions

So after I've trained and tested my models these are the results of the accuracies for the different models.





Logistic Regression	98%
Decision Trees	99%
SGD Square	62%
SGD Log_loss	98%
SGD Hinge	99%
SGD Modified_huber	99%
Gradient Boosting	99%

Passive Aggressive	99%
SVM	99%
KNN	74%
Naive Bayes	94%
RandomForest	98%

In conclusion, our analysis of various machine learning models highlights significant variations in accuracy results. Logistic Regression, Decision Trees, and several SGD variants consistently demonstrated high accuracy rates, reaching up to 99%, showcasing their robust performance in classifying the dataset. On the contrary, SGD Square and KNN exhibited lower accuracy percentages, suggesting potential challenges in handling specific dataset characteristics. Further exploration into the unique strengths and weaknesses of each model could guide targeted improvements. Notably, the consistently high performance of models like Decision Trees and SVM at 99% accuracy underscores their reliability for complex classification tasks. Looking ahead, incorporating neural network methods, such as LSTM, presents an intriguing avenue for enhancing classification accuracy and understanding temporal patterns in the data.

Overall, our findings provide insights into model performance and open doors for refining approaches in future research.

References

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

https://thecleverprogrammer.com/2021/02/10/passive-aggressive-classifier-in-machine-learning/

https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/

https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/

https://www.capitalone.com/tech/machine-learning/what-is-logistic-regression/

https://medium.com/capital-one-tech/naives-bayes-classifiers-for-machine-learning-2e548bfbd4a1

https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/

https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb

https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26

https://scikit-learn.org/stable/modules/sgd.html#:~:text=Stochastic%20Gradient%20Descent%20(SGD)%20is,Vector%20Machines%20and%20Logistic%20Regression.

https://scikit-learn.org/stable/modules/svm.html

https://datagy.io/python-support-vector-machines/

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/