UNIVERISTATEA BABEȘ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ ENGLEZĂ

LUCRARE DE LICENŢĂ

Detectarea stirilor false si evaluarea credibilității stirilor prin intermediul Machine Learning

Conducător științific
Lect. Dr. Bădărînză Ioan

Absolvent
Bîrligă Alexandru-Robert

2022

BABEȘ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE IN ENGLISH

DIPLOMA THESIS

Fake News Detection and News Reliability Evaluation by Means of Machine Learning

Supervisor
Lect. Dr. Bădărînză Ioan

Author
Bîrligă Alexandru-Robert

2022

Contents

1 Introduction

2 Core Concepts of Machine Learning & Natural Language Processing

Machine learning and natural language processing are important components laying at the foundation of this thesis. Therefore, the current chapter provides an overview of some of the relevant fundamental concepts, required for understanding the thesis further on.

2.1 Introduction

Artificial intelligence (AI), originally coined as a term and founded as an academic discipline in the 1950s [1], has become a buzzword not only in the domain of computer science, but also in the popular culture. The growth in popularity and general interest is owed to the accelerated technological advancements and exponential increase in the volumes of data, which fueled the progression of AI from mostly bare theory to a gradual actuality.

There is no singular universally accepted definition for artificial intelligence. This term generated a considerable amount of differences and confusion among specialists[2]. There are numerous proposed definitions, but one that encapsulates to a certain extent a common essence encountered in a fair part of them could be formulated as follows:

Definition 1 *Artificial intelligence is a branch of computer science, concerned with developing computer programs that approach problems emulating the human model of thinking and its typical processes, such as learning, reasoning and self-correction.*

Another term coined in the 1950s is "machine learning". Machine learning is a subfield of artificial intelligence, based on the idea that computers can be programmed to "learn" from data and tackle tasks with minimal human intervention. A definition for this field of study, based on Tom Mitchell's proposal [3], is the following one:

Definition 2 *Machine learning studies the capability of a computer program to learn from experience E with respect to some task T and some performance measure P, improving with experience E its performance on the task T, measured by P.*

This definition is a more mathematical one, operating with three variables:

1. E (experience) stands for the dataset collected for the learning process, which is leveraged for extracting a general formula from a set of particular examples.

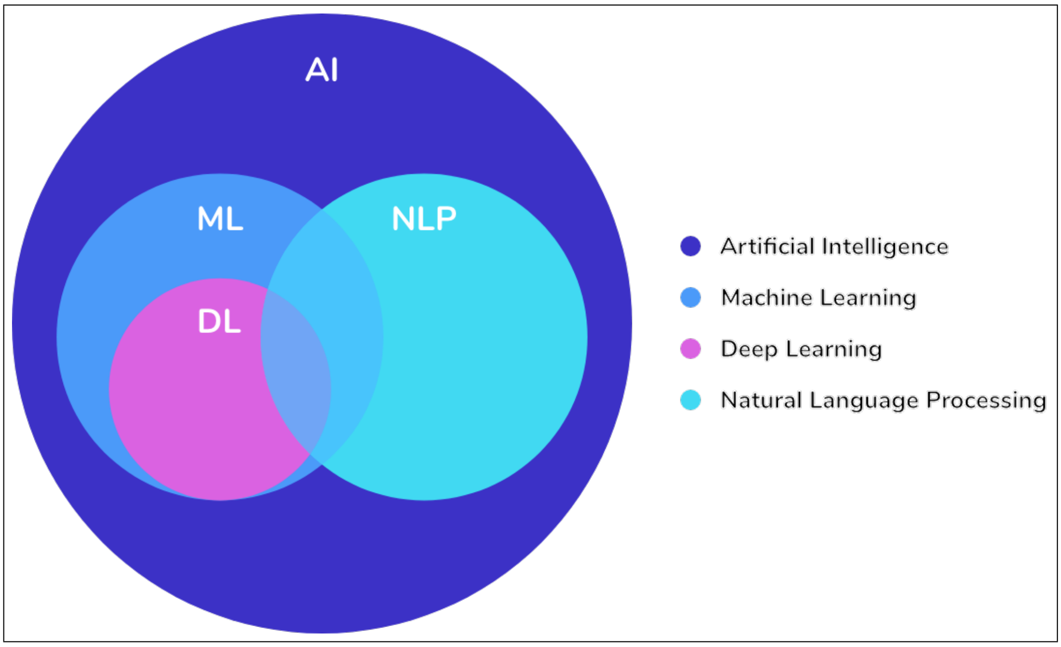
2. T (task) represents the problem which is desired to be solved via a machine learning algorithm.

3. P (performance measure) refers to a set of metrics applied on the machine learning model to mathematically quantify its performance.

Machine learning is a novel field which has been successfully implemented and notably pushed the boundaries in many real applications, including pattern recognition, speech recognition, audio processing, natural language processing (e.g. language translation, search results, smart assistants, text analysis) and computer vision (e.g. object detection, image classification, segmentation).

Out of these, natural language processing has probably the most applications in the area of fake news detection, the subject of this thesis.

Definition 3 *Natural language processing (NLP) is a field of study at the intersection of artificial intelligence and linguistics, which is comprised of a range of computational techniques for analyzing and representing human language [4].*



2.2 More on Machine Learning

2.2.1 Core Concepts of Supervised Learning

Machine learning algorithms can be divided into three categories: **supervised learning**, **unsupervised learning** and **reinforcement learning**. Supervised learning algorithms are trained on a set of input-output pairs, with the aim of creating a mathematical model to map any general input to an output. Conversely, unsupervised learning techniques deal with unlabelled data from which patterns are derived, through various clustering and association algorithms. Lastly, reinforcement learning alg involves algorithms learning by being rewarded for desired behaviors and punished for undesired ones.

In the context of fake news detection, the majority of research revolves around supervised learning algorithms, hence that is what this thesis will further focus upon.

Supervised learning operates with annotated (labeled) datasets during the training and testing process. Every element from the training dataset is called a feature vector, in which each dimension (i.e. feature) characterizes the input data from a certain aspect. The goal is to learn the correlation between the set of features and the labels for every training example, so that through generalization, the label of a potentially unmet input can be predicted. This process of identifying correlations, anomalies and patterns within a dataset is known as **data mining**.

There are generally no perfect models that can generate the correct output with 100% accuracy for all of the inputs. The discrepancies between the ground truth labels and the actual predictions of the model can be expressed by a **loss function**. Thereby, the purpose of the supervised learning algorithm can also be understood as the minimization of the loss function, which translates into the maximization of the accuracy and other performance metrics of the model.

Supervised learning algorithms can be split into two types: **classifications** and **regressions**. Both algorithms are used for making predictions, but they differ primarily in the type of the output produced. For a given input, classification algorithms predict discrete values, called classes or categories, from a finite set of values, to which the feature vectors are considered to belong. Contrarily, regression algorithms predict continuous values (integer or floating point values), which usually denote either quantities or sizes. For instance, let us consider weather prediction based on certain parameters, such as: humidity, temperature or atmospheric pressure. This task could be approached in two ways: predict if the weather is hot or cold (binary classification) or predict a precise value for the temperature, such as 24 degrees Celsius (regression).

As for fake news detection, in order to determine whether an article is fake or real, a classification algorithm is most suitable. The algorithms employed within this project are probabilistic classifiers, which are not only able to make a categorical prediction (i.e. fake or real), but they can also compute the probability distribution of a news article over the set of classes. Probabilities are more revealing than mere binary values, because they measure the confidence with which the classifier predicts a certain class.

Within this project, the following classification algorithms were used for the problem of fake news detection: **Support Vector Machine**, **Decision Tree**, **Logistic Regression**, **K-nearest Neighbors**.

2.2.2 K-nearest Neighbors

K-nearest Neighbors (KNN) is one of the simplest machine learning supervised algorithms, which is also known to be quite versatile and well-performing. It functions on the principle of proximity relative to the previous training examples. It can be used as a regression algorithm, but it is most encountered in classification problems. KNN is **non-parametric** and **lazy-learning** or **slow-learning**, the two primary properties of KNN.

By non-parametric, it is meant that the classifier does not make any prior strong assumptions about the underlying structure of the data by deciding beforehand the number of parameters of the mapping function which the algorithm models during training. Opposed to parametric algorithms, which can discard the training data after the model is build, KNN stores each training example as a parameter and makes all the upcoming predictions based on the k most similar training patterns performed earlier on. Practically, the sole assumption the model makes is the correlation or similarity between the feature vectors analyzed previously and the new training examples.

By lazy-learning or slow-learning, it is meant that KNN is technically not having an explicit training process and it does not learn from the training data immediately. During the training phase, the KNN algorithm simply stores in memory each of the feature vectors from the training dataset, relying on observable data similarities and distance metrics.

The KNN algorithm can be outlined by the following steps which are iteratively performed for each training or testing example:

1. Locate and store the feature vector on the graph, as a data point.

2. Calculate the distance between the current data point and the rest of the previous training examples. In order to calculate the distance, various formulas can be used, such as the Euclidean distance, the Manhattan Distance, the Minkowski distance or the cosine distance.

3. Consider the k nearest neighbor data point and count the frequency of each label.

4. Assign a class label to the current data point according to the majority vote rule: the label that is in the greatest proportion present around the data point, from the k considered closes ones.

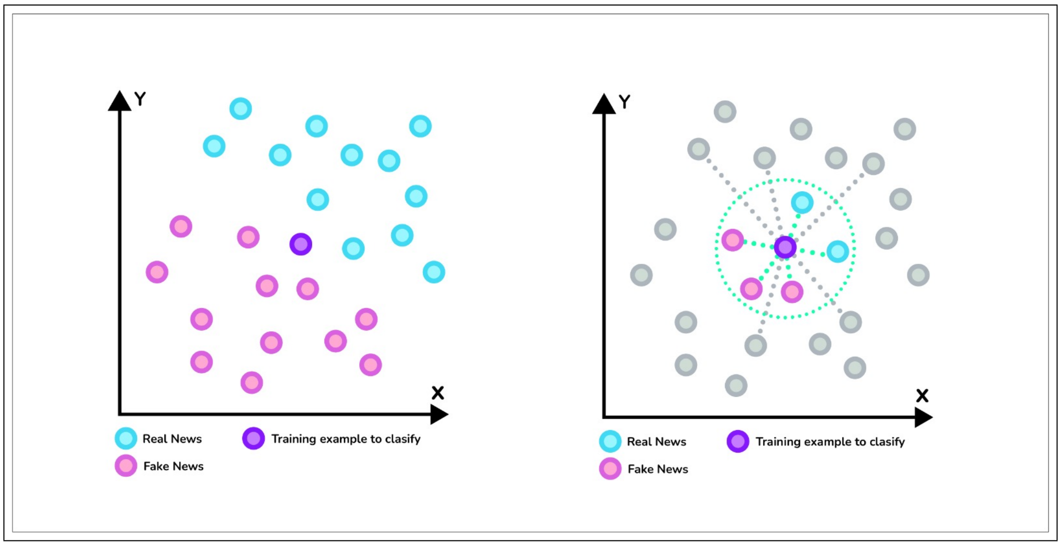


Figure 1: KNN algorithm selecting the K nearest neighbors.

The **advantages** of KNN is that it is relatively simple to implement, can learn non-linear decision boundaries, no training time, new data can be added anytime without a negative effect on the accuracy and there is only one hyperparameter to handle (i.e. k). As of **disadvantages**, does not work very efficiently on large datasets and it is sensitive to noisy data.

2.2.3 Decision Tree

Decision Tree is a supervised learning algorithm used for both classification and regression, having a tree-like structure, where the leaves correspond to output classes and the sub-branches represent series of decision rules, each leading finally to the prediction of an output class. The sub-branches are comprised of internal nodes, each of them containing different features of the dataset. Decision trees can be viewed as graphical representation covering all the solutions to a problem based on a specified set of conditions.

It is called a decision tree because comparable to a tree, it starts from the root node (representing the entire dataset) and expands to an increasing number of sub-branches gradually, learning from the past training examples. Each internal leaf stands for a decision rule. By decision rule, we mean a simple IF-THEN statement, consisting of a condition and a prediction. For instance, IF a person is 65 years old and is obese, THEN they have a high probability of hearth attack of diabetes.

There are two main operation performed on a decision tree, namely splitting (i.e. the operation of ramifying the root node or a decision node into two sub-branches) and pruning (i.e. the operation of compression of the size of the tree, by elimination unimportant or redundant sections of the tree). Also, by pruning one can reduce the complexity and overfitting from the tree.

A decision tree can be decomposed in the following steps:

1. Initialize the tree with only the root node, which contains the whole dataset.

2. Use an attribute selection measure to split the node into two sub-branches according to the best attribute.

3. Create a new decision node containing the best attributed identified at the prior step.

4. Recursively apply the first 3 steps to all the resulting subtrees.

5. Stop the recursivity as soon as the nodes cannot be classified further anymore and regard the final nodes as leaves.

The pruning process mentioned above can be done in two ways: post-pruning (i.e. execute the pruning after the construction of the decision tree is completed, technique particularly used when the depth of the tree is large and shows signs of overfitting) or pre-prunning (i.e. execute the prunning some time before the completion of the decision tree, technique useful to prevent overfitting).

On the one hand, the upsides of decision tree include: less data preparation is needed (not critical to standardize and normalize the data), data scaling is not required and it has the ability to capture non-linear relationships. On the other hand, the downsides are that decision tree generally require more computational resources and execution time

2.2.4 Logistic Regression

Logistic regression is a supervised machine learning algorithm which is used for classifications and involves drawing a hypothesis function which best fits the categories/classes of a dataset.

Opposed to the linear regression, where the aim is to find a linear function, logistic regression looks for other types of function, such as the commonly utilized sigmoid/logistic function. The sigmoid function is a S-shaped curve, useful for binary classification tasks, given that the domain of the function can be the whole real values set and the output is either greater than and very close to 0 or less than and very close to 1. The sigmoid function "squishes" any value towards the top and bottom margins. If the output of the sigmoid function is less than 0.5, the algorithm will predict that the data point belongs to first class, otherwise that it belongs to second class. The equation of the sigmoid function is the following one:

While in linear regression algorithms, adjusting the linear function is done by minimizing the sum of the distances between each data point to the line (i.e. the least squares method), in logistic regression the maximum likelihood estimation is used instead. The maximum likelihood estimation method attempts to maximize the conditional probability of observing the data X given a specific probability distribution and its parameters theta [5].

The advantages of this algorithm include: efficient training process, easy and intuitive implementation, ability to easily extend to multiple classes. Conversely, there is also one major disadvantage: the constraint of linear boundaries.

2.2.5 Support Vector Machine

Support Vector Machine is another popular supervised learning algorithms, that is often preferred for the high accuracy along with low computation power. The main idea of the algorithm is to search for a hyperplane which separates the data points into the possible output classes.

From all the possible hyperplanes meeting this requirement, the algorithm has to specifically select that which is furthest from the data points of each category. Maximizing the margin between the hyperplane and the data points results in a more solid decision boundary between the classes, making the predictions more confident and reliable. If the hyperplane is discrepantly closer to the data points of a certain class and further from another class, then the observation located in the close proximity of the hyperplane could be classified in the group which is considerably further from.

The name of "Support Vector Machine" reveals the essential elements that the algorithms is based upon: support vectors (i.e. data points that are closest to the hyperplane and which influence the position and orientation of the hyperplane).

2.3 More on Natural Language Processing

Natural Language Processing (often abbreviated as NLP) is a sub-field of computer science, artificial intelligence and linguistics, concerned with tasks involving human language processing. Given that this thesis explores linguistic-based or content-based fake news detection, there are several NLP concepts and core terminology critical for understanding the machine learning algorithms that are to be presented in the following chapters.

Pre-processing is a critical initial step when training an NLP model. The purpose of this general practice is to prepare the dataset for training, by applying a series of operation on text data.

2.4 Fake News

Fake news is not simply a product of the last decades. It has been part of humanity for at least centuries. Its first significant outburst dates back to 1400s, when the printing press was invented. Ever since then, fabricated news have been more and more weaponized to manipulate people at increasingly larger scales. In 1835, The New York Sun published 6 articles presenting the discovery of life on the moon. Both World Wars were heavily marked by propaganda, and false and misleading news [6]. And more recently, the US 2016 presidential elections and the COVID-19 pandemic were subjects of large amounts of fake news.

Despite being so widespread, there is no globally agreed definition of the "fake news" term. Nonetheless, many sources provide definitions which have two primary elements in common: authenticity and intent. As for authenticity, fake news contain claims that are factually and verifiably false. As for intent, the major purpose of fake news is to deceive the consumers [7]. That being said, a concise definition of fake news could be formulated in the following manner:

Definition 4 *By fake news we mean any news item that is deliberately and factually untrue and misleading.*

2.5 Fake News Detection

Fake News Detection represents a new area of study. On the one hand, it emerged thanks to the continuous advancements in Artificial Intelligence and, specifically, Deep Learning. On the other hand, it appeared as virtually a necessity, given the recent escalation of the fake news phenomenon, on account of the simultaneous raise of digital media.

Digital and social media are environments in which the dissemination of information is more facile and rapid than at any point in history. Fabricated news have been heavily published and shared as a means of deception, on subjects of high impact, such as the 2016 US presidential elections and the COVID-19 pandemic.

At present, there are multiple fake news identification resources. They each contribute in various manners to validating the veracity of news and debunking rumours, hoaxes, conspiracies.

2.6 Related Work

3 Application Development

This chapter provides a complete rundown on the development process of the fake news detection application build for this thesis, from inception to completion.

3.1 Functionalities

The application developed for this thesis is a web extension built for Google Chrome, which provides users with an accessible tool for evaluating the reliability of online news. The extension performs a multifaceted analysis of any given news article, based upon its URL, headline, content and authors, by means of machine learning algorithms, web scraping and crowdsourcing.

The key functionalities of the application are the following:

1. upon landing on an online news article from a known news source, automatically extract some critical information about the article, based on its HTML source code (i.e. URL, headline, content and authors)

2. after successfully extracting the required information about the article, automatically initiate a multifaceted reliability analysis, which rates the current article from the following standpoints:

   (a) the biased/deceptive character of the language used in the content of the article, predicted by one of the machine learning algorithms implemented for content-based fake news detection

   (b) the deceptive/sensationalist character of the language used in the headline of the article, predicted by one of the machine learning algorithm implemented for title-based clickbait detection

   (c) the reliability of the source based upon the list of reliable / perennial sources web-scraped from Wikipedia [8]

   (d) the reliability of the source based upon previous user feedback

   (e) the credibility of the authors based upon previous user feedback

3. after the completion of the analysis, the users have the ability to provide their own rating / feedback for the article

3.2 Architecture and Technologies

3.2.1 Database

3.2.2 Front-end

* chrome extension * React * state management (why I tried with redux, but just used chrome.storage.local instead) * talk about what chrome extesion APIs I used * explain what is a service worker * explain what is a popup.js * explain using react hooks, components * some npm packages I used

The **client-side** (**front-end**) of the application was built as a Chrome Extension, so that all the fake news detection and reliability analysis functionalities are immediately accessible upon opening a given news article on the browser. Compared to a standard web front-end application, a Chrome extension (and browser extensions in general) have a more peculiar architecture.

Chrome extensions have three main components:

1. **Service Worker**, a JavaScript script running in the background of the browser, reacting to events emitted from the browser, such as tabs/windows being open/closed, refreshing the webpage, modifying the URL from a tab etc. Compared to popup and content scripts, the lifecycle of the service worker is not dependent of any webpage. Therefore, the service worker can be running across webpages, terminate only after becoming idle and restart only when needed (i.e. when having to handle an event from the browser). Service workers are also useful if one wants to store a temporary local state shared across tabs or run an action in the background which should not run only while the popup of the extension is open.

2. **Popup**, the user interface of the chrome extension, which gets toggled when clicking on the icon of the extension located on the taskbar. By means of HTML, CSS and JavaScript, an interface comparable to an ordinary webpage can be written, with certain implicit limitations, such as lack of routing, cannot use standard Redux for state management etc. The lifecycle of the popup lasts only while the popup is open, so if one wants to preserve state between popup sessions, one should either choose to store state in the

background script, localStorage or Chrome's Sync or Local Storage.

3. **Content Script**, JavaScript scripts running in the context of webpages. They are useful when it comes to reading or modifying the Document Object Model (DOM) of the webpage. Even though they can perform changes to the HTML and CSS source code of the webpage, they are unable to interact with the JavaScript of the webpage (i.e. access and use functions or variables defined in the context of the web page or extension). Also, they cannot access the Chrome APIs and events. Despite having these limitations, which many have as workaround to communicate via a messages protocol with parent extension, content scripts are useful in many scenarios.

The popup of the news reliability extension was written using the JavaScript library called React, which provides a more efficient development environment and functionalities for writing component-based web interfaces. With the help of components,

3.2.3 Back-end

* I have 2 servers: Python (w/ Flask) and Node.js (w/ Express) * What my Python server does * What my Node.js server does * Why I decided to split into two servers * Talk about design architectures, OOP etc. * Talk about the ai models, training (briefly, as I am going to develop on this further when talking about the algorithm) * some dependcies / packages I used (sklearn, panda, newspaper, chirio)

3.2.4 Other tools

* git and Github for source control * Insomnia as API client * Adobe XD for UI / UX * Notion for task management

3.3 Analysis standpoints

3.3.1 Article content-based detection

3.3.2 Heading clickbaitness

3.3.3 Source reliability

3.3.4 Crowdsourcing / User Feedback

4 Conclusion and Future Work

5 References

[1] Matthew Helm, Andrew Swiergosz, Heather Haeberle, Jaret Karnuta, Jonathan Schaffer, Viktor Krebs, Andrew Spitzer, and Prem Ramkumar. Machine learning and artificial intelligence: Definitions, applications, and future directions. *Current Reviews in Musculoskeletal Medicine*, 13, 02 2020.

[2] Joost N Kok, Egbert J Boers, Walter A Kosters, Peter Van der Putten, and Mannes Poel. Artificial intelligence: definition, trends, techniques, and cases. *Artificial intelligence*, 1:270–299, 2009.

[3] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[4] Elizabeth D. Liddy. Natural language processing. 2001.

[5] Jason Brownlee. A gentle introduction to logistic regression with maximum likelihood estimation. `https://machinelearningmastery.com/logistic-regression-with-maximum-likelihood-estimation/`. Accessed: 2022-08-20.

[6] Posetti J. and Matthews A. A short guide to the history of 'fake news' and disinformation. 2018.

[7] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. 2017.

[8] Wikipedia:reliable sources/perennial sources. `https://en.wikipedia.org/wiki/Wikipedia:Reliable_sources/Perennial_sources`. Accessed: 2022-08-20.

[9] Ahmed H, Traore I, and Saad S. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 2017.

[10] Choudhary A. and Arora A. Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications*, 2020.

[11] Li Deng and Dong Yu. Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7(3–4):197–387, jun 2014.

[12] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

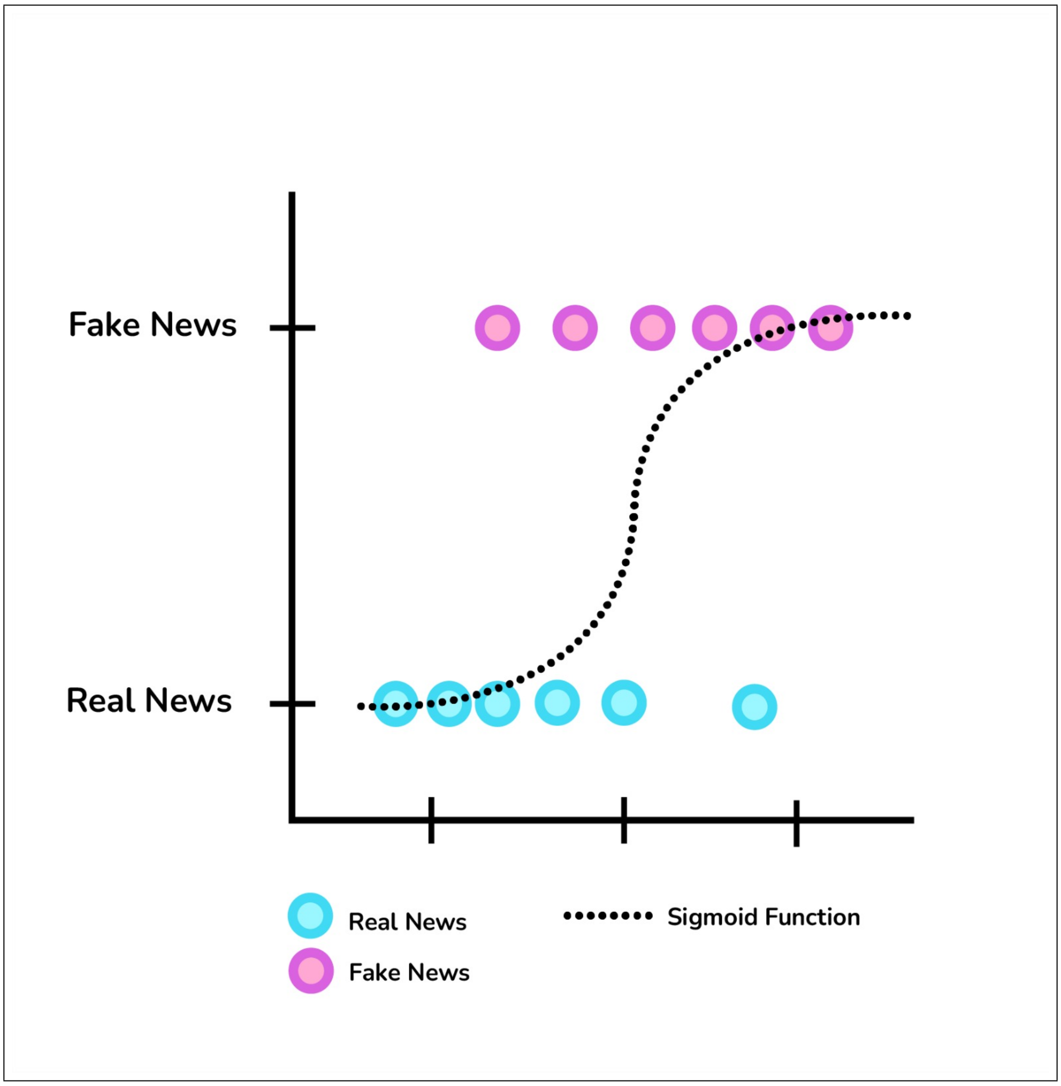


Figure 2: KNN algorithm selecting the K nearest neighbors.

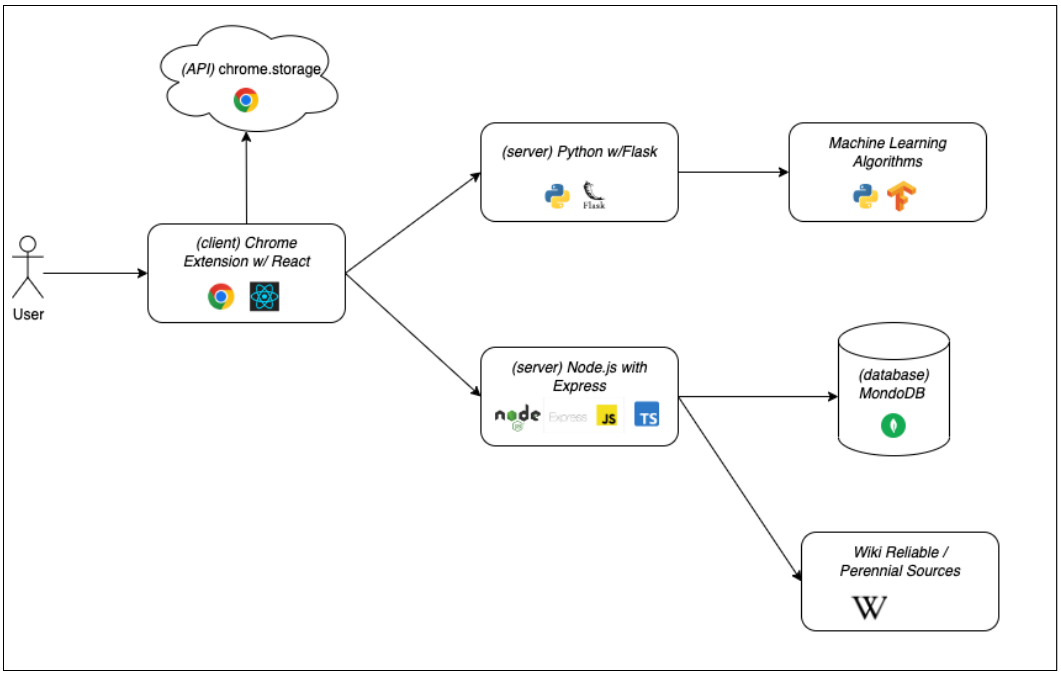

(API) chrome.storage
User
(client) Chrome Extension w/ React
(server) Python w/Flask
Flask
Machine Learning Algorithms
(server) Node.js with Express
node
Express
JS
TS
(database) MondoDB
Wiki Reliable / Perennial Sources
W

Figure 3: Diagram depicting the architecture of the application.