**Content :**

|  |  |
| --- | --- |
| * Introduction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
| * Why Python is the preferred language for Machine Learning? . . . | 6 |
| * Machine Learning and Cyber Security. . . . . . . . . . . . . . . . . . . . . | 7 |
| * Machine Learning models and Algorithms. . . . . . . . . . . . . . . . . | 8 |
| * Spam Email detection using Machine Learning. . . . . . . . . . . . . . | 11 |
| * Turning the Spam Email Classifier into a Web Application. . . . . . . | 12 |

**Abstract**

Currently, machine learning techniques are some of the hottest trends in information technology. They impact on every aspect of our lives, and they affect every industry and field. Machine learning is a cyber weapon for information security professionals. It is now a necessary aspect of every modern project. Combining mathematics and cutting-edge optimization techniques and tools can provide amazing results. Applying machine learning and analytics to information security is a step forward in defending against advanced real-world attacks and threats.

Here we will be building Spam detector using the NLTK. The principle of this type of classifier is simple; we need to detect the words used by spammers. We are going to build a spam/non-spam binary classifier using Python and the nltk library, to detect whether or not an email is spam.

**Introduction :**

**Python**

Python is a widely used high-level programming language for general-purpose programming. Apart from being open source programming language, python is a great object-oriented, interpreted, and interactive programming language. Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems. New built-in modules are easily written in C or C++ (or other languages, depending on the chosen implementation). Python is also usable as an extension language for applications written in other languages that need easy-to-use scripting or automation interfaces.

**Machine learning**

Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. In today’s world every system that does well has a machine learning algorithm at its heart. Take for example Google Search engine, Amazon Product recommendations, LinkedIn, Facebook etc, all these systems have machine learning algorithms embedded in their systems in one form or the other. They are efficiently utilising data collected from various channels which helps them get a bigger picture of what they are doing and what they should do. Machine learning is so pervasive today that we probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI.

**Why Python is the preferred language for Machine Learning?**

Python is widely considered as the preferred language for teaching and learning Ml (Machine Learning). Few simple reasons are:

* It’s simple to learn. As compared to c, c++ and Java the syntax is simpler and Python also consists of a lot of code libraries for ease of use.
* Though it is slower than some of the other languages, the data handling capacity is great.
* Open Source! – Python along with R is gaining momentum and popularity in the Analytics domain since both of these languages are open source.
* Capability of interacting with almost all the third party languages and platforms.

**Machine Learning and Cyber Security**

Data has posed perhaps the single greatest challenge in cyber security over the past decade. For a human, or even a large team of humans, the amount of data produced daily on a global scale is unimaginable. Thankfully, this is where machines excel, automating simple tasks such as processing and classification to ensure analysts are left with a manageable quantity of actionable insights. ML is routinely used in cyber security for the following purposes:

* **Pattern recognition** — Identifying phishing emails based on content or sender info, identifying malware, etc.
* **Anomaly detection** — Spotting unusual activity, data, or processes (e.g., fraud detection for online banking or gambling).
* **Natural language processing (NLP)** — Converting unstructured text such as a webpage into structured intelligence.
* **Predictive analytics** — Processing data and identifying patterns in order to make predictions and identify outliers.

Hackers are always trying to use new, sophisticated techniques to attack modern organizations. Thus, as security professionals, we need to keep ourselves updated and deploy the required safeguards to protect assets. Many researchers have shown thousands of proposals to build defensive systems based on machine learning techniques. For example, the following are some information security models:

* **Supervised learning**:
  + Network traffic profiling
  + Spam filtering
  + Malware detection
* **Semi-supervised learning**:
  + Network anomaly detection
  + C2 detection
* **Unsupervised learning**:
  + User behavior analytics
  + Insider threat detection
  + Malware family identification

**Machine Learning models and Algorithms**

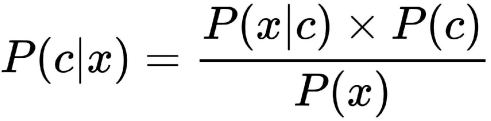
In order to teach machines how to solve a large number of problems by themselves, we need to consider the different machine learning models. As we know, we need to feed the model with data; that is why machine learning models are divided, based on datasets entered (input), into four major categories: supervised learning, semi-supervised learning, unsupervised learning, and reinforcement. In this section, we are going to describe each model in a detailed way, in addition to exploring the most well-known algorithms used in every machine learning model. Before building machine learning systems, we need to know how things work underneath the surface.

1. **Supervised**

We talk about supervised machine learning when we have both the input variables and the output variables. In this case, we need to map the function (or pattern) between the two parties. The following are some of the most often used supervised machine learning algorithms.

1. **Bayesian classifiers**

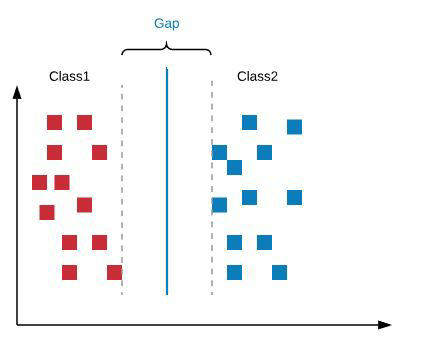
According to the *Cambridge English Dictionary*, bias is the action of supporting or opposing a particular person or thing in an unfair way, allowing personal opinions to influence your judgment. Bayesian machine learning refers to having a prior belief, and updating it later by using data. Mathematically, it is based on the Bayes formula:

****

One of the simplest Bayesian problems is randomly tossing a coin and trying to predict whether the output will be heads or tails. That is why we can identify Bayesian methodology as being probabilistic. Naive Bayes is very useful when we are using a small amount of data.

1. **Support vector machines**

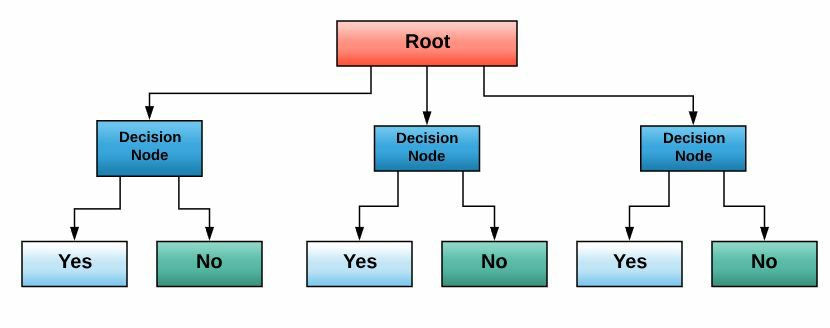
A support vector machine (SVM) is a supervised machine learning model that works by identifying a hyperplane between represented data. The data can be represented in a multidimensional space. Thus, SVMs are widely used in classification models. In an SVM, the hyperplane that best separates the different classes will be used. In some cases, when we have different hyperplanes that separate different classes, identification of the correct one will be performed thanks to something called a margin, or a gap. The margin is the nearest distance between the hyperplanes and the data positions. You can take a look at the following representation to check for the margin:



The hyperplane with the highest gap will be selected. If we choose the hyperplane with the shortest margin, we might face misclassification problems later.

1. **Decision trees**

Decision trees are supervised learning algorithms used in decision making by representing data as trees upside-down with their roots at the top. The following is a graphical representation of a decision tree:



Data is represented thanks to the Iterative Dichotomiser 3 algorithm. Decision trees used in classification and regression problems are called CARTs. They were introduced by Leo Breiman.

1. **Semi-supervised**

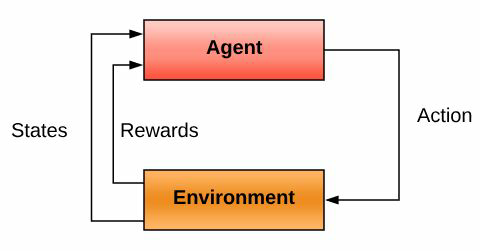
Semi-supervised learning is an area between the two previously discussed models. In other words, if we are in a situation where you are using a small amount of labeled data in addition to unlabeled data, then we are performing semi-supervised learning. Semi-supervised learning is widely used in real-world applications, such as speech analysis, protein sequence classification, and web content classification. There are many semi-supervised methods, including generative models, low-density separation, and graph-based methods (discrete Markov Random Fields, manifold regularization, and mincut).

1. **Unsupervised**

In unsupervised learning, we don't have clear information about the output of the models. The following are some well-known unsupervised machine learning algorithms.

1. **Reinforcement**

In the reinforcement machine learning model, the agent is in interaction with its environment, so it learns from experience, by collecting data during the process; the goal is optimizing what we call a long term **reward**. We can view it as a game with a scoring system. The following graph illustrates a reinforcement model:



**Spam Email Detection Using Machine Learning**

Email spam refers to the use of an email system to send unsolicited emails especially advertising emails to a group of recipients. Unsolicited emails mean the recipient did not grant permission for receiving those emails.

The use of spam emails has been growing in popularity since the last decade and is a problem faced by most email users. Email IDs of users who receive email spam are usually obtained by spam bots (automated software that crawls the internet for email addresses).

Email spam is still a problem even today, and spammers still approach it the spam way. Spam accounts for billions of emails sent every day which makes up 98% of all emails. Spam causes businesses billions of dollars every year.

Even though antivirus software has come a long way, infected PCs, malwares are still the major sources of spam. There are billions of public IPs available for use; each one could have thousands of PCs behind it including potentially infected malwares

Email spam is also termed as junk email, these are suspicious messages sent in bulk through emails. Most of the email spam messages are commercial in nature. They contain links that look genuine and convincingly familiar however the links leads to phishing websites that hosts malware.

* Since we’ll be working with text data, we’ll be using the Python-based library Natural Language Toolkit ([NLTK](http://www.nltk.org/)), which has rich functionality in natural language processing tasks.
* Here we will use **Naive Bayes classifier**, which is a simple yet powerful classification algorithm that has been widely applied to spam filtering [before](https://github.com/shanbady/NLTK-Boston-Python-Meetup).

The classifier tries to choose the most probable class, or label, among the two classes, spam and ham, i.e. Untitledbased on what it has learned about the features (presence or frequency of words in the emails of each type). More precisely, it’s trying to choose the most probable class given the words in the e-mail.

**Spam detection algorithm will consist of 5 steps-**

**Load the data**

-load spam emails

-load ham emails

-shuffle

**Extract the features**

-remove stopwords

-count or register word occcurences

**Preprocess**

-tokenise

-lamentise

-make lowercase

**Train the classifier**

-split into train and test sets.

-apply a classifier

**Evaluate**

-accuracy on the train and test sets

-information features

**Performance of Spam Detection Model :**

**Total Mails : 5857 mail**

**Total Features : 5857**

**Training = 4685 emails**

**Testing = 1172 emails**

**Training Accuracy is 0.9974386339381003**

**Testing Accuracy is 0.9914675767918089**

**Most Informative Features**

**cc = 1 ham : spam = 230.6 : 1.0**

**vince = 1 ham : spam = 204.3 : 1.0**

**privacy = 1 spam : ham = 90.4 : 1.0**

**php = 1 spam : ham = 86.5 : 1.0**

**713 = 1 ham : spam = 69.9 : 1.0**

**artwork = 1 spam : ham = 69.1 : 1.0**

**forwarded = 1 ham : spam = 68.5 : 1.0**

**identity = 1 spam : ham = 67.1 : 1.0**

**8859 = 1 spam : ham = 67.1 : 1.0**

**castillo = 1 spam : ham = 59.4 : 1.0**

**norton = 1 spam : ham = 59.3 : 1.0**

**advertisement = 1 spam : ham = 55.4 : 1.0**

**853 = 1 ham : spam = 55.2 : 1.0**

**2005 = 1 spam : ham = 52.7 : 1.0**

**macromedia = 1 spam : ham = 51.6 : 1.0**

**2004 = 2 spam : ham = 51.5 : 1.0**

**satisfaction = 1 spam : ham = 49.6 : 1.0**

**ect = 1 ham : spam = 47.3 : 1.0**

**houston = 1 ham : spam = 46.1 : 1.0**

**pharmaceutical = 1 spam : ham = 45.7 : 1.0**

**Turning the Spam Email Classifier into a Web Application**

**Objective:**-Having prepared the code for classifying emails previously, we will develop a web application that consists of a simple web page with a form field that lets us enter an email. After submitting the email to the web application, it will render it on a new page which gives us a result of **spam or not spam.**

**Technology and Tools**

* Python3
* Front end: HTML, CSS, Jinja
* Back end: Flask

**Tools Description:-**

* **Flask:**- Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications. This API allows us to utilize the predictive capabilities through HTTP requests.

* **Jinja:**- Flask comes packaged with the powerful Jinja templating language

Templating languages essentially contain variables as well as some programming logic, which when evaluated (or rendered into HTML) are replaced with **actual values.** The variables and/or logic are placed between tags or delimiters. For example, Jinja templates use {% ... %} for expressions or logic (like for loops), while {{ ... }} is used for outputting the results of an expression or a variable to the end user. The latter tag, when rendered, is replaced with a value or values, and is seen by the end user.

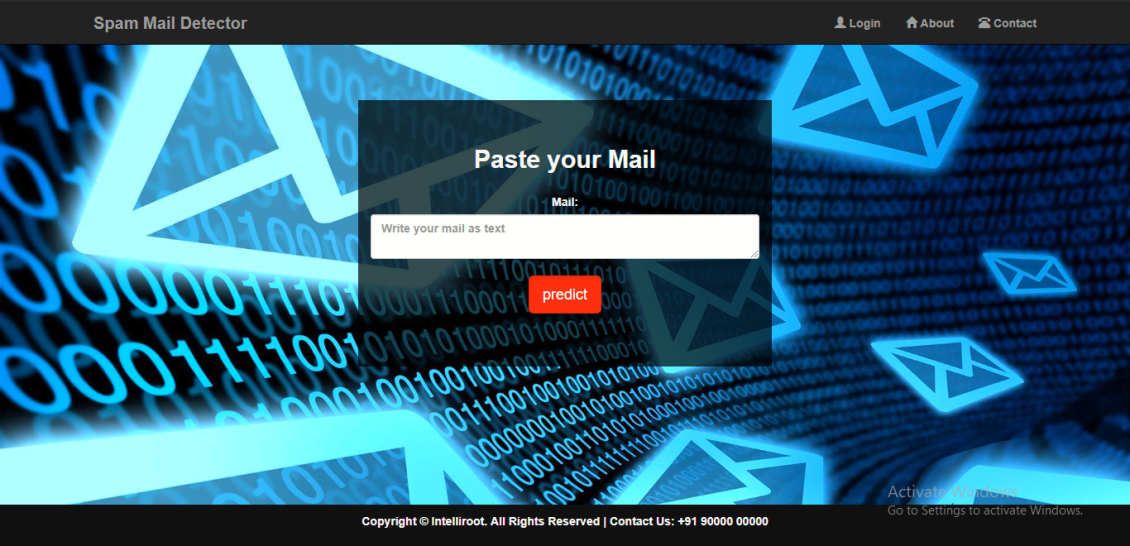
* **HTML:-** HTML stands for Hypertext Markup Language.

Hypertext is ordinary text that has been dressed up with extra features, such as formatting, images, multimedia, and links to other documents. Markup is the process of taking ordinary text and adding extra symbols. Each of the symbols used for markup in HTML is a command that tells a browser how to display the text.

* **CSS:-** Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, we can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

**Modules:-**

* Email submission module- Here user can paste any of the copied mails in order to check for spam. After submitting the mail user is redirected to result page.



* Result Module- This page displays the concerned result i.e. spam or ham.

