Intrusion Detection Model Using Machine Learning Algorithm On Big Data Environment

A mini project report submitted toJawaharlalNehruTechnologicalUniversity

inpartial fulfillmentoftherequirementsfortheawardofdegreeof

BACHELOROFTECHNOLOGY

In

COMPUTERSCIENCEANDENGINEERING

SubmittedBy

M Ruthvik Mohan-18P71A0578

UndertheGuidanceof

Mr.Manish Kumar Sinha

DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING

SWAMIVIVEKANANDAINSTITUTEOFTECHNOLOGY

(ApprovedbyAICTE&AffiliatedtoJawaharlalNehruTechnologicalUniversity,Hyderabad)

Mahbub College Campus, Patny Centre,
SecunderabadYearofsubmission: 2020-2021

SwamiVivekanandaInstituteofTechnology

(ApprovedbyAICTE&AffiliatedtoJawaharlalNehruTechnologicalUniversity,Hyderabad)

MahbubCollegeCampus,Patny Centre,Secunderabad

COMPUTERSCIENCEANDENGINEERING

CERTIFICATE

This is to certify that the Industry oriented mini project entitled “**Intrusion Detection Model Using Machine Learning Algorithm On Big Data Environment**”Submittedby **M Ruthvik Mohan-18P71A0578.**Inpartialfulfilment forthe award of the Degree of Bachelor of Technology in Computer Scienceand Engineering to the Jawaharlal Nehru Technology University is a record of bonafideworkcarried out by himunder my guidance andsupervision.

InternalGuide **HeadoftheDept.**

Mr.Manish Kumar Sinha Dr.J.Manoranjani

ExternalExaminer

Intrusion Detection Model Using Machine Learning Algorithm On Big Data Environment

ABSTRACT:

An Intrusion Detection Model (IDM) using a Machine Learning (ML) algorithm on a Big Data environment is a method for identifying and preventing unauthorized access to a computer system. The IDM utilizes a ML algorithm to analyze large sets of data, or "Big Data," in order to identify patterns and anomalies that may indicate a security breach. These patterns and anomalies are then used to create a model that can detect intrusions in real-time. The use of a Big Data environment allows for the IDM to process and analyze large amounts of data quickly and accurately, making it more effective at detecting and preventing intrusions. This approach can be used in a variety of industries, including finance, healthcare, and government, to improve the security of sensitive informationis compared to the Chi Logistic Regression classifier and the results show that the Spark Chi SVM model has a high performance, reduces training time and is efficient for handling Big Data.

EXISTING SYSTEM:

There are several existing systems that utilize an Intrusion Detection Model (IDM) using a Machine Learning (ML) algorithm on a Big Data environment. One example is the Apache Mahout project, which is an open-source framework for creating scalable ML algorithms. It includes an implementation of the Random Forest algorithm, which can be used for intrusion detection in a Big Data environment. Another example is the Hadoop-based Distributed File System (HDFS) and MapReduce framework, which can be used to store and process large amounts of data. Overall, these existing systems demonstrate the capability of IDM using ML algorithms on big data environment to effectively detect and prevent intrusions by analyzing large amounts of data..

PROPOSED METHOD:

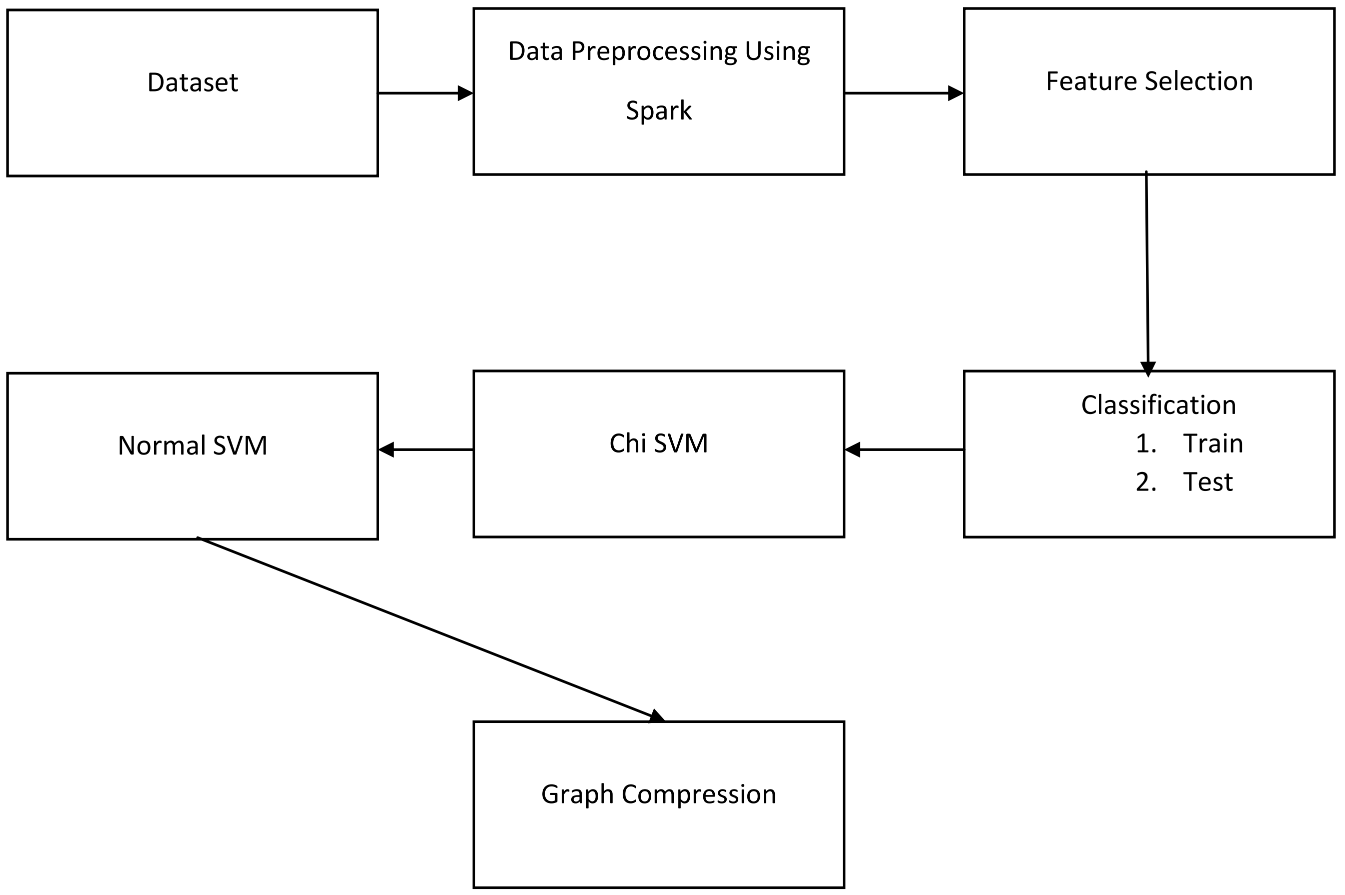
The proposed model in this study is the Spark Chi SVM model, as illustrated in Figure 1. The following steps are included in this model:

1. Loading the dataset and converting it into Resilient Distributed Datasets (RDD) and Data Frame in Apache Spark.
2. Data preprocessing to prepare the data for analysis.
3. Feature selection to identify relevant features for the analysis.
4. Training the Spark Chi SVM model using the training dataset.
5. Testing and evaluating the model using the KDD dataset.

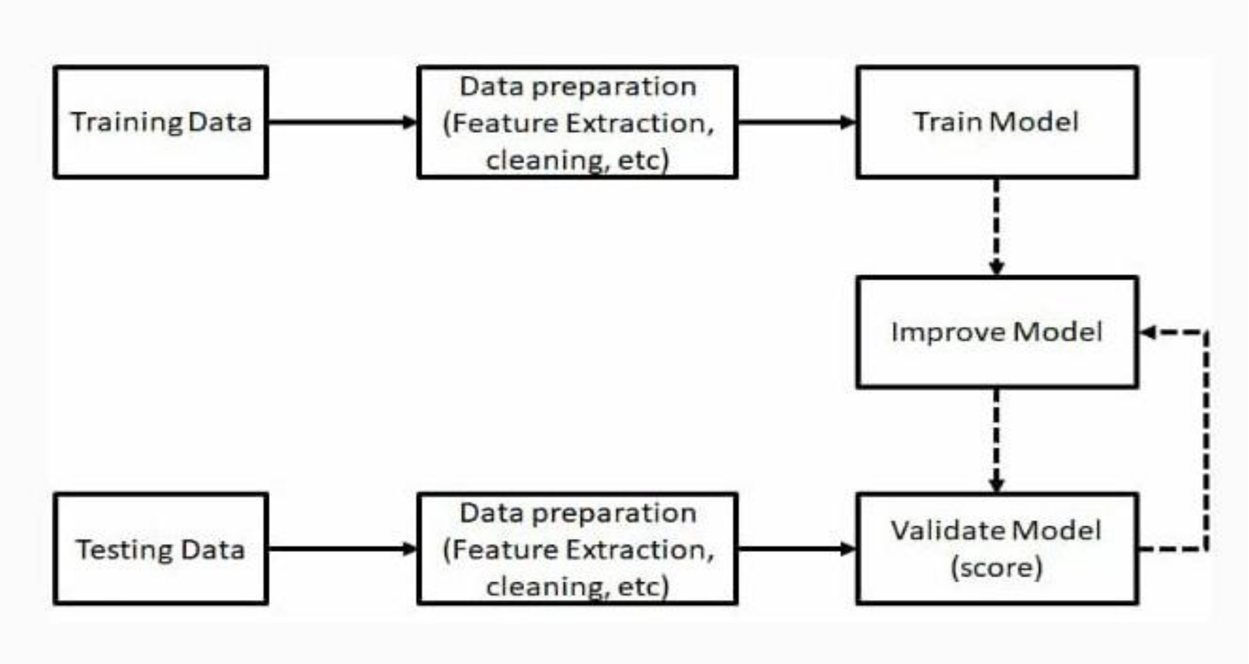
DATASET DESCRIPTION:

The proposed model in this study is evaluated using the KDD99 dataset. The dataset includes 494,021 instances. The KDD99 dataset includes 41 attributes, with a "class" attribute that indicates whether a given instance is normal or an attack. Table 1 provides a description of the KDD99 dataset attributes and class labels.

SYSTEM ARCHITECTURE:



TECHNICAL ARCHITECTURE:


Training Data
Data preparation (Feature Extraction, cleaning, etc)
Train Model
Improve Model
Testing Data
Data preparation (Feature Extraction, cleaning, etc)
Validate Model (score)

MODULES:

Dataset and Pre-processing:

We used a dataset that includes both fake and genuine profiles for our study. The dataset includes various attributes such as the number of friends, followers, and status count. Data for training and testing are separated from the dataset. Classification algorithms are trained using the training dataset, and the testing dataset is used to determine the efficiency of the algorithm. For this study, 80% of both genuine and fake profiles were used to create the training dataset, and 20% of both profiles were used to create the testing dataset.

Feature Selection:

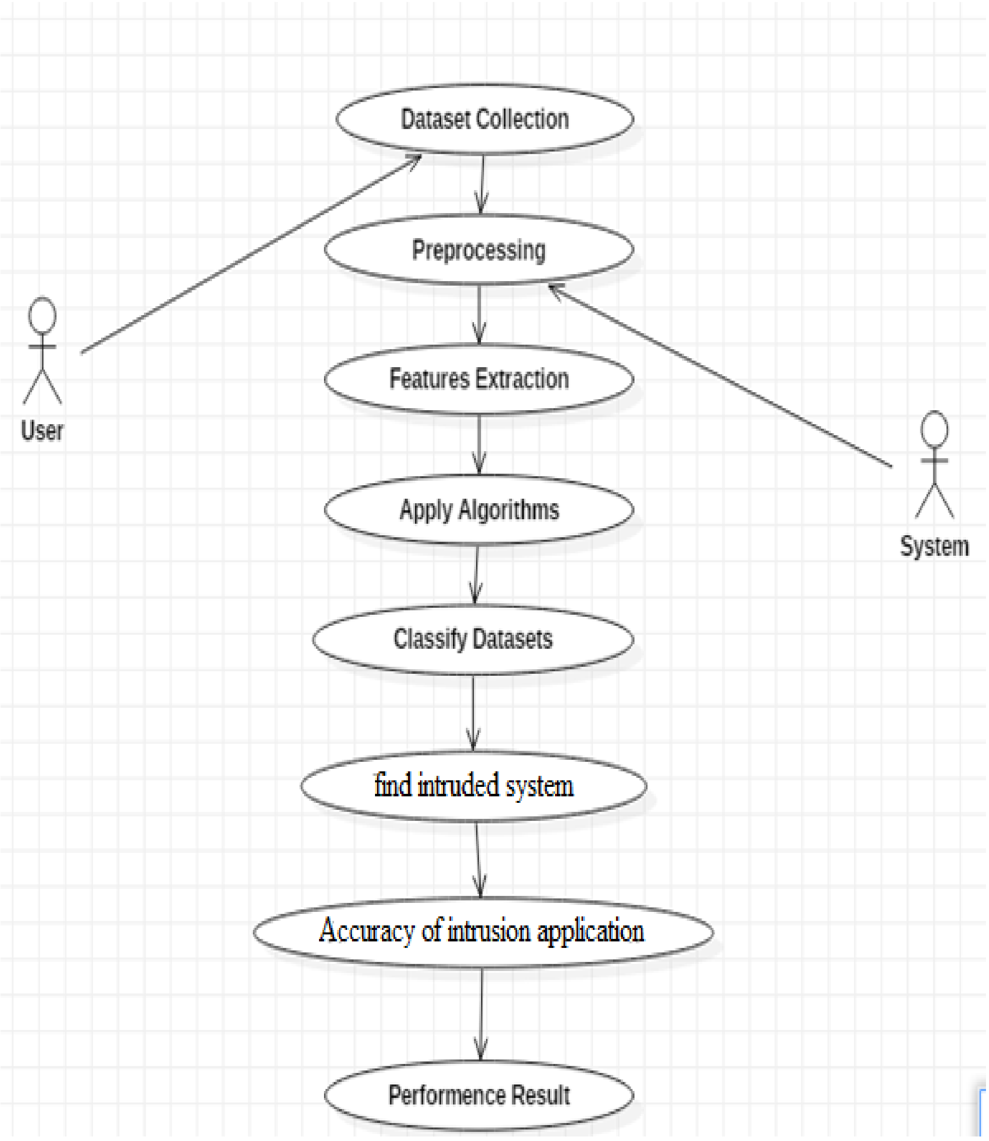
In this study, features are selected to apply classification algorithms. The classification algorithm is discussed further in the paper. Attributes are chosen as features if they are independent of other attributes and they increase the efficiency of the classification. The specific features chosen for this study are discussed in further detail.

After the selection of attributes, a dataset of profiles that have already been classified as fake or genuine is needed for training the classification algorithm. We used a publicly available dataset of 1337 fake users and 1481 genuine users, which includes various attributes such as name, status count, number of friends, followers count, favorites and languages known.

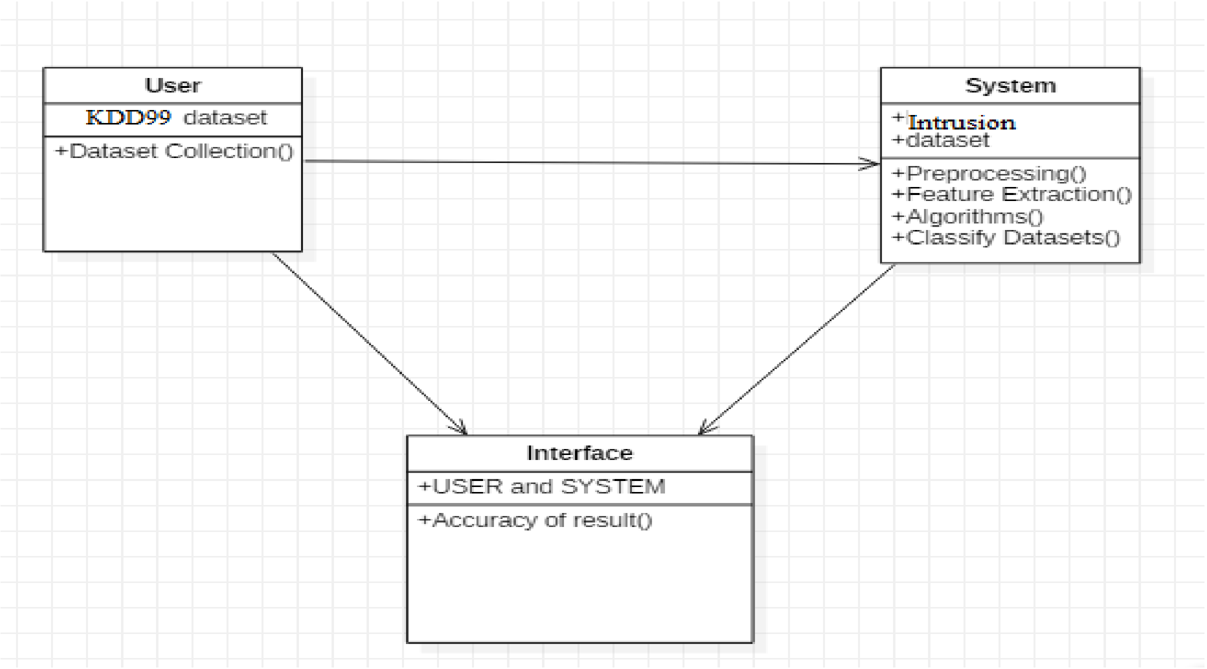
Classification:

Classification is the process of categorizing a data object into categories, called classes, based on the features or attributes associated with that data object. Classification uses a classifier, an algorithm that processes the attributes of each data object and outputs a class based on that information. In this project, we use Support Vector Machine (SVM) as a classifier. SVM is an elegant and robust technique for classification on large datasets similar to those found in social networks with several millions of profiles.

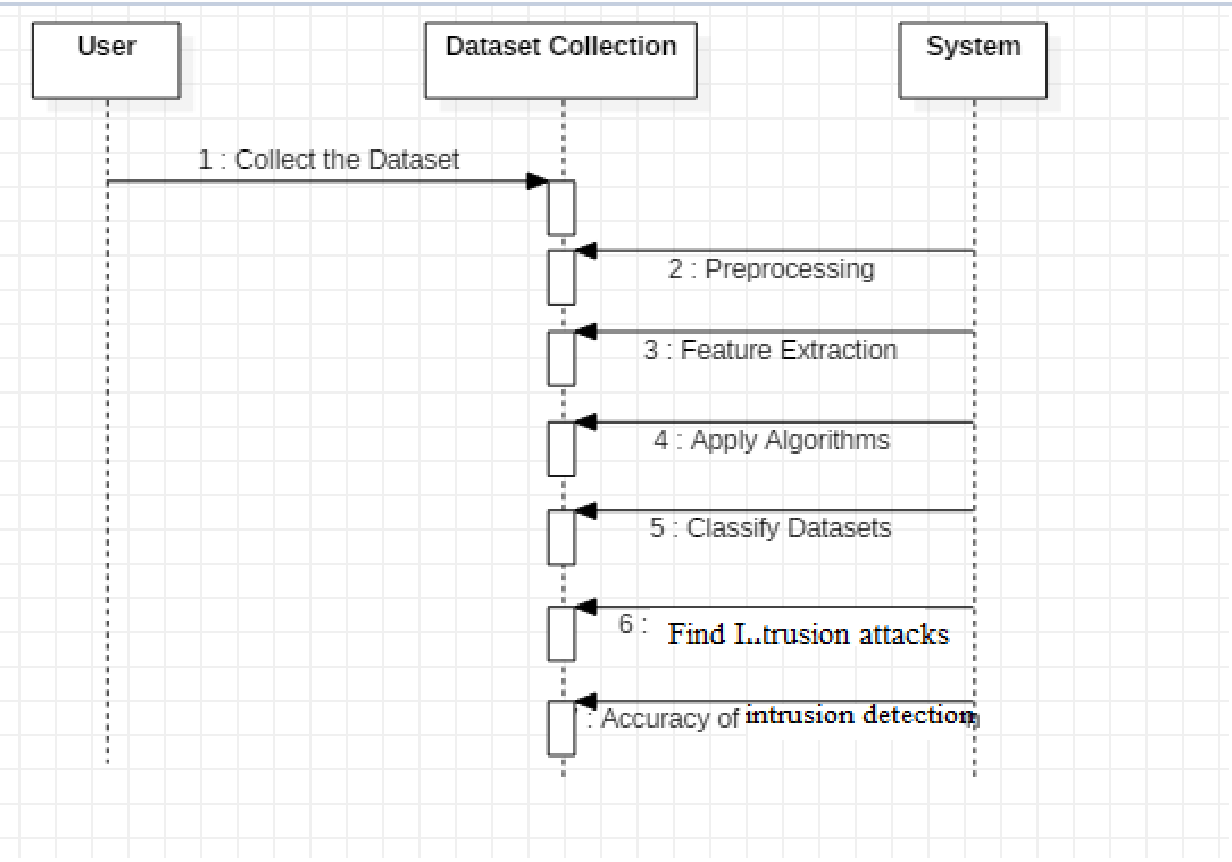
USECASE DIAGRAM:



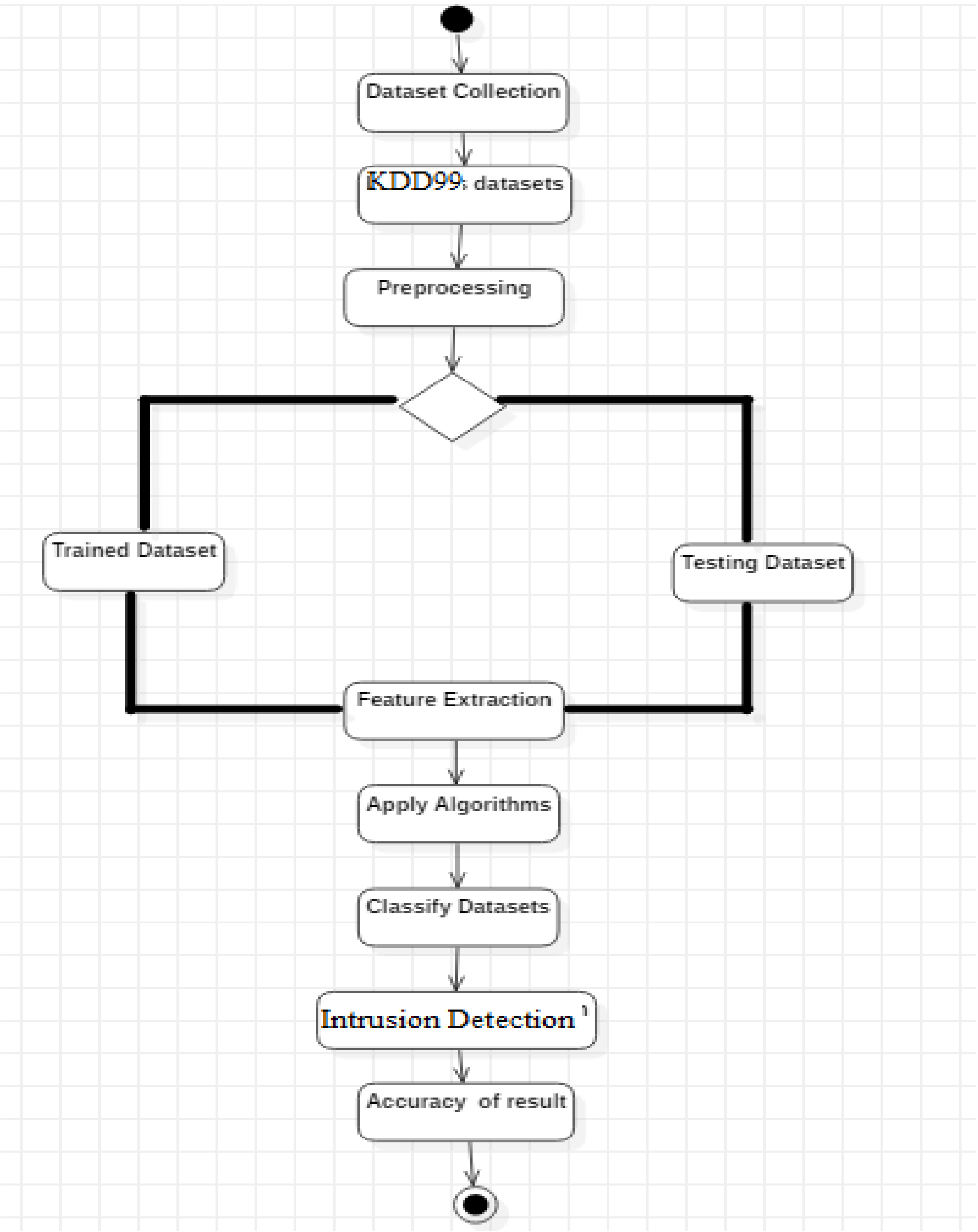
CLASS DIAGRAM:



SEQUENCE DIAGRAM:



ACTIVITY DIAGRAM:



ALGORITHM

RANDOM FOREST

Random Forest is a type of supervised machine learning algorithm that is based on ensemble learning. Ensemble learning is a method of combining multiple algorithms or instances of the same algorithm to create a more powerful prediction model. The Random Forest algorithm combines multiple decision trees, resulting in a "forest" of trees, hence the name "Random Forest." This algorithm can be used for both regression and classification tasks.

HOW RANDOM FOREST WORKS

The Random Forest algorithm works by following these basic steps:

1. Pick N random records from the dataset.

2. Build a decision tree based on these N records.

3. Repeat steps 1 and 2 for the number of trees specified in the algorithm.

When the algorithm is applied to a classification problem, each tree in the forest predicts the category to which the new record belongs. The new record is then assigned to the category that wins the majority vote.

ADVANTAGES OF USING RANDOM FOREST

There are several advantages to using the Random Forest algorithm for classification and regression:

1. The Random Forest algorithm is not biased because it relies on the collective "intelligence" of multiple decision trees, each trained on a subset of data. This reduces the overall bias of the algorithm.

2. The algorithm is very stable, meaning that even if new data is introduced, it is unlikely to significantly impact the overall algorithm.

3. The Random Forest algorithm works well with both categorical and numerical features.

4. In addition, Random Forest also provides three common data pre-processing steps:

- **Formatting:** The data may be in a format that is not suitable for the analysis, and it may need to be converted to a different format.

- **Cleaning:** Cleaning data involves removing or fixing missing data and anonymizing or removing sensitive information.

- **Sampling:** It is possible to take a smaller representative sample of the data to work with if the dataset is too large.

By using these preprocessing steps, the algorithm can be more efficient and accurate in dealing with large data.

5. Random Forest Algorithm also provides feature importance, which is useful in identifying the most important features from the dataset which can be used to reduce the dimensionality of the dataset.

DOMAIN SPECIFICATION:

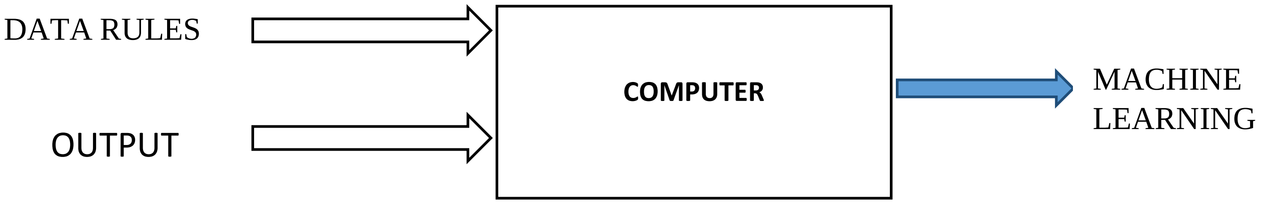
MACHINE LEARNING:

Machine learning is a system that can learn from examples through self-improvement, without being explicitly programmed by a developer. The key innovation of machine learning is that a machine can learn from data, leading to accurate results. Machine learning combines data with statistical tools to predict an output that can be used by organizations to make actionable insights. It is closely related to data mining and Bayesian predictive modeling. The machine takes in data as input and uses an algorithm to generate answers.

A common application of machine learning is providing recommendations, such as movie or series recommendations on Netflix which are based on the user's historical data. Tech companies are also using unsupervised learning to improve the user experience through personalization. Machine learning is also used for a wide range of tasks such as fraud detection, predictive maintenance, portfolio optimization, and automating tasks.

Machine Learning vs. Traditional Programming

Traditional programming is vastly different from machine learning. In traditional programming, a programmer codes all the rules based on consultation with an expert in the field for which the software is being developed. Each rule is built on a logical foundation, and the machine will execute an output following the logical statement. However, as the system becomes more complex, more rules need to be written, which can become difficult to maintain. Machine learning, on the other hand, allows the system to learn and improve on its own without the need for explicit programming of rules.

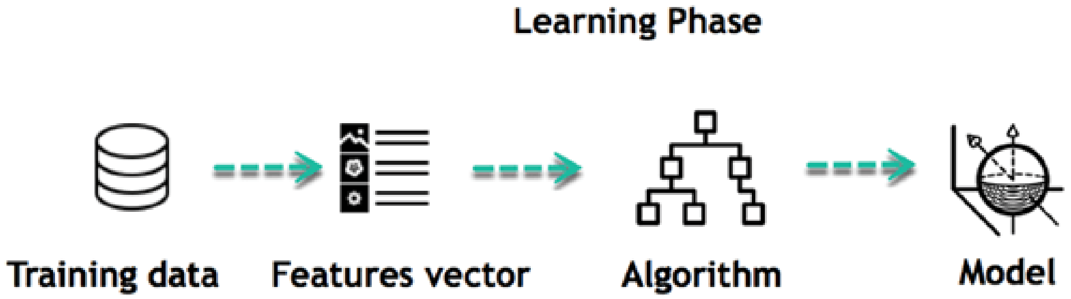


How does Machine learning work?

Machine learning is the process in which the machine learns to make predictions or take actions. It works similarly to how humans learn, through experience. As humans gain more knowledge, they can make more accurate predictions. Similarly, when a machine is given examples, it can figure out the outcome. However, if it is given a previously unseen example, the machine may have difficulty making a prediction.

The core objective of machine learning is learning and inference. The machine learns by discovering patterns in data. A crucial part of the process is carefully selecting the data to provide to the machine. The list of attributes used to solve a problem is called a feature vector, which can be thought of as a subset of data that is used to tackle a problem.

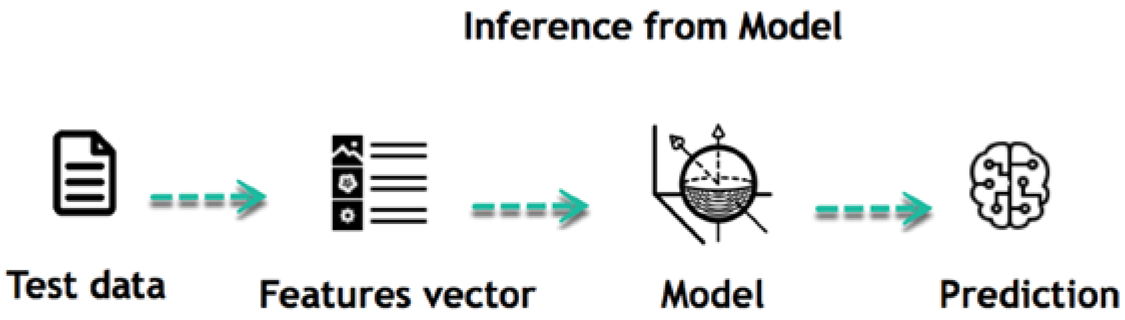
The machine uses algorithms to simplify reality and transform the discovered patterns into a model. The learning stage is used to describe the data and summarize it into a model. This model is then used to make predictions or take action in new situations.



For example, let's say the machine is trying to understand the relationship between an individual's wage and their likelihood of going to a fancy restaurant. After analyzing the data, the machine discovers a positive relationship between wages and going to a high-end restaurant. This relationship is the model. The machine can then use this model to predict whether someone with a certain wage is likely to go to a fancy restaurant or not.

INFERRING:

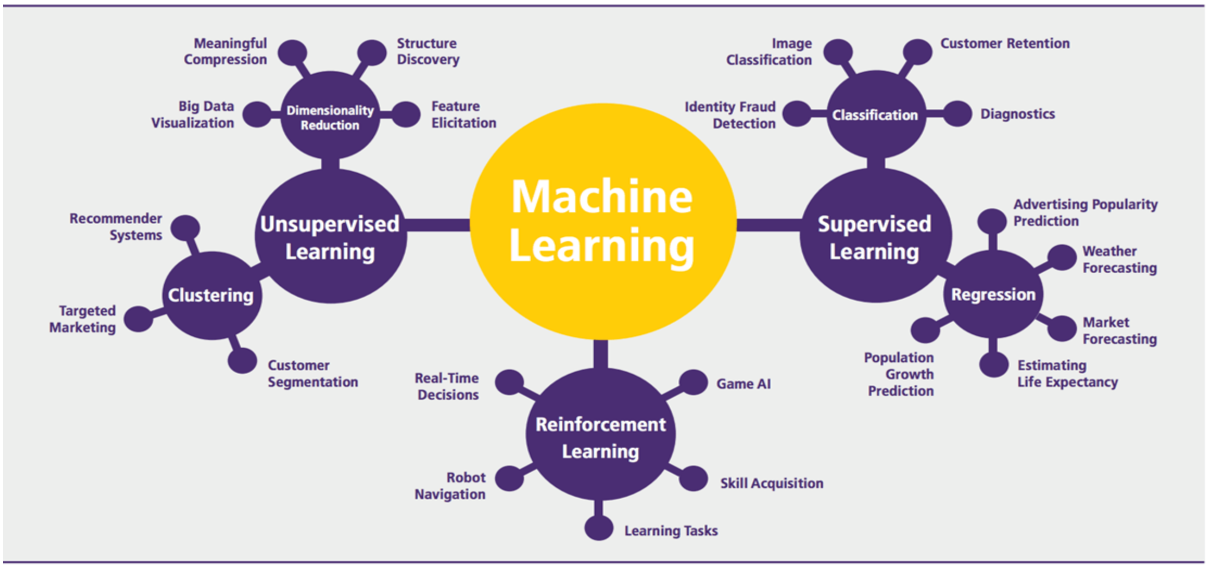
Once the model is built, it can be tested on new, unseen data. The new data is transformed into a feature vector, processed through the model, and a prediction is made. This is a powerful aspect of machine learning. There is no need to update the rules or retrain the model. The previously trained model can be used to make inferences on new data.



The life cycle of Machine Learning programs can be summarized in the following steps:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the algorithm
6. Collect feedback
7. Refine the algorithm
8. Repeat steps 4-7 until the results are satisfactory
9. Use the model to make predictions

Once the algorithm becomes proficient at drawing the correct conclusions, it can apply that knowledge to new sets of data.Machine learning algorithms are used in various fields such as image and speech recognition, natural language processing, fraud detection, predictivemaintenance, and more.



Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

SUPERVISED LEARNING:

In supervised learning, the algorithm uses training data and feedback from humans to learn the relationship between given inputs and a given output. For example, a practitioner can use marketing expenses and weather forecasts as input data to predict the sales of cans. This type of learning is used when the output data is known and the algorithm is used to predict new data.

UNSUPERVISED LEARNING:

In unsupervised learning, the algorithm is not given any labeled data and must find patterns and relationships in the input data on its own. Clustering and dimensionality reduction are examples of unsupervised learning tasks.

There are two types of **supervised** learning:

CLASSIFICATION:

In classification tasks, the algorithm is trained to assign a label or category to new data based on the input features. For example, to predict the gender of a customer for a commercial, data such as height, weight, job, salary, and purchasing basket are collected. The algorithm is trained to recognize the relationship between these features and the label (i.e., male or female) and can then be used to make predictions on new data. The label can have two or more classes.

REGRESSION:

In regression tasks, the algorithm is trained to predict a continuous value instead of a label. For example, predicting the price of a house based on its size, location, and number of rooms.

Algorithm Name	Description	Type
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable 3 is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Regression Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. The SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.	Regression (not very common) Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees are a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommender system	Help to define the relevant data for making a recommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

APPLICATION OF MACHINE LEARNING:

Machine learning and artificial intelligence are being used in many industries to improve efficiency, reduce errors, and gain insights from data. In the finance industry, ML is used to detect patterns and prevent fraud. Government organizations use ML for public safety and utility management. The healthcare industry has been using ML for image detection for some time. In marketing, AI is used to optimize customer relationships and marketing campaigns. As the use of machine learning and AI continues to grow, the benefits for various industries will become increasingly apparent.In addition to these applications, machine learning is also being used in various other industries and fields such as transportation, retail, agriculture, and energy. For example, in transportation, machine learning is being used to optimize logistics and supply chain management, while in retail it is being used for personalized marketing and product recommendations. In agriculture, machine learning is being used for crop monitoring and yield prediction, and in energy, it is being used for smart grid management and energy consumption prediction. Overall, the applications of machine learning are vast and diverse, and its potential for improving efficiency and decision-making across industries is significant.

Example of application of Machine Learning in Supply Chain

Machine learning is a valuable tool for supply chain management as it can improve efficiency and accuracy in various processes. One example is the use of visual pattern recognition for physical inspection and maintenance in the logistics hub. Unsupervised learning can be used to quickly analyze diverse data sets and identify patterns, allowing for efficient and accurate quality inspections. IBM's Watson platform, for instance, can analyze visual and systems-based data to detect damage in shipping containers and provide real-time recommendations. Additionally, machine learning can also be used to improve inventory forecasting, leading to potential cost savings and increased sales.

Example of Machine Learning Google Car

This is a great example of how machine learning can be applied in the field of autonomous vehicles. The Google car uses a combination of sensors and algorithms to gather data about its environment and make decisions about how to navigate safely. The use of machine learning allows the car to learn from past experiences and improve its driving skills over time. Additionally, the ability to process large amounts of data in real time enables the car to make quick and accurate decisions in dynamic and unpredictable situations. The use of machine learning in autonomous vehicles is expected to revolutionize the way we think about transportation and has the potential to significantly improve safety on the roads.

Deep Learning:

The neural network is a set of layers, each one with a specific role. The first layers learn the most basic features of the data, such as edges and shapes. As the data flows through the network, each layer learns more complex features. The last layer is called the output layer and is responsible for making predictions. Deep learning is particularly useful for tasks that involve image and speech recognition, natural language processing, and video analysis. It has been used in a wide range of applications such as self-driving cars, image and voice recognition, and medical diagnosis. Due to a large amount of data and computational power required, deep learning is often implemented on powerful hardware such as GPUs.

Reinforcement Learning:

Reinforcement learning is a type of machine learning where systems are trained by receiving virtual "rewards" or "punishments" based on their actions, learning through trial and error. This type of learning has been used in various applications such as gaming, where AI-powered agents can improve their gameplay through reinforcement. Some popular algorithms in reinforcement learning include Q-learning, Deep Q networks, SARSA, and DDPG.

One example of the application of reinforcement learning is in the financial industry, where AI is being used to improve credit scoring and risk assessment. Companies like Underwrite.ai use deep learning to predict which loan applicants are more likely to pay back loans, resulting in more accurate and efficient lending decisions.

Another example is in the human resources industry, where companies like Under Armour use AI-powered solutions to streamline their hiring process, reducing the time to fill open positions and improving the quality of hires.

In the marketing industry, AI is being used to improve customer service management and personalization. For example, companies are using deep learning to analyze customer audio and assess their emotional tone, allowing for real-time adjustments to the customer service experience. AI is also being used to analyze customer data and provide personalized marketing campaigns.

	Machine Learning	Deep Learning
Data Dependencies	Excellent performances on a small/medium dataset	Excellent performance on a big dataset
Hardware dependencies	Work on a low-end machine.	Requires a powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication
Feature engineering	Need to understand the features that represent the data	No need to understand the best feature that represents the data
Execution time	From a few minutes to hours	Up to weeks. Neural Network needs to compute a significant number of weights
Interpretability	Some algorithms are easy to interpret (logistic, decision tree), some are almost	Difficult to impossible

impossible (SVM, XGBoost)

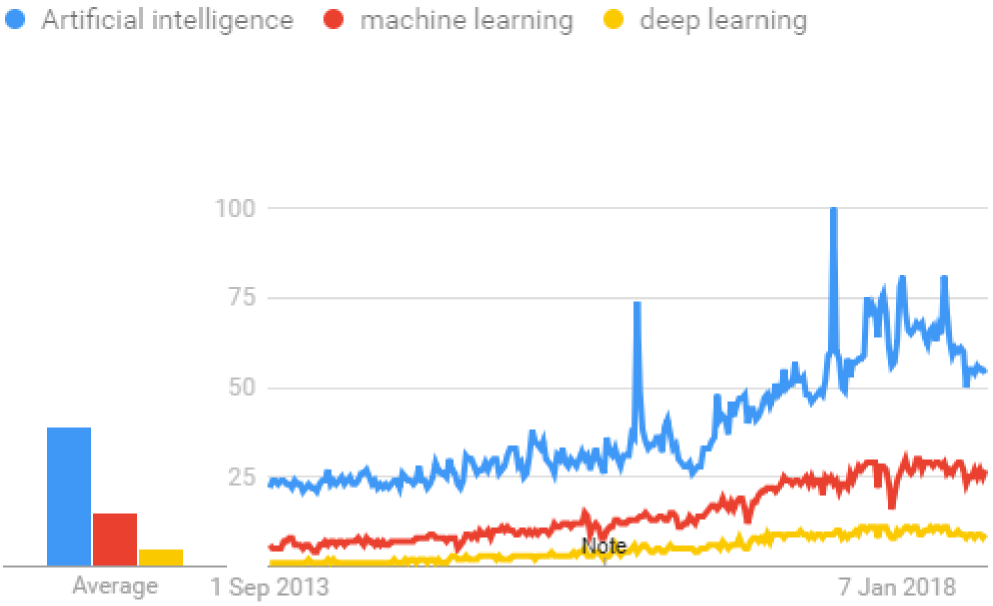
Difference between Machine Learning and Deep Learning:

When to use ML or DL?

In the table below, we summarize the difference between machine learning and deep learning.

	Machine learning	Deep learning
Training dataset	Small	Large
Choose features	Yes	No
Number of algorithms	Many	Few
Training time	Short	Long

Machine learning is also less computationally intensive than deep learning, making it more accessible for smaller companies and organizations. Additionally, machine learning models are often more interpretable than deep learning models, making it easier to understand how the model arrived at its predictions. On the other hand, deep learning models are more adeptat handling complex and unstructured data, such as images and audio. Overall, the choice between machine learning and deep learning depends on the specific task and the available resources.



TensorFlow:

TensorFlow is an open-source deep-learning library developed by Google Brain Team. It is designed to accelerate machine learning and deep neural network research and is widely used by researchers, data scientists, and programmers. TensorFlow takes input as a multi-dimensional array or tensor and allows users to construct a flowchart of operations (called a Graph) that they want to perform on that input. The input goes in at one end, flows through the system of operations, and comes out the other end as output.

TensorFlow is designed to run on multiple CPUs or GPUs and even mobile operating systems and has wrappers in several languages such as Python, C++, and Java. It can be used for both the development and run phases of machine learning projects. During the development phase, training is usually done on a desktop or laptop, while the run or inference phase can be done on a variety of platforms such as desktop computers, cloud services, or mobile devices.

TensorFlow also includes a feature called TensorBoard which enables users to monitor and visualize what TensorFlow is doing. This makes it easy to understand and troubleshoot the model. TensorFlow supports a wide range of algorithms such as linear regression, classification, deep learning classification, boosted tree regression, and classification.

REQUIREMENTS ANALYSIS:

SOFTWARE REQUIREMENTS:

- Python
- Anaconda Navigator

Python built-in modules

- NumPy
- Pandas
- Matplotlib
- SkLearn
- Seaborn

ANACONDA NAVIGATOR:

Anaconda Navigator is a desktop graphical user interface (GUI) supplied with the Anaconda distribution that allows you to run programs and manage conda packages, environments, and channels without having to use command-line commands. Both Anaconda Cloud and a local Anaconda Repository are searchable by Navigator. It works with Windows, Mac OS, and Linux.

Additionally, Navigator also allows you to launch Jupyter Notebooks, JupyterLab, and other popular applications such as Rstudio, Spyder, and Visual Studio Code. It also includes built-in support for managing Python and R environments, making it easy to switch between different versions of these languages. Navigator also provides an easy way to launch and manage virtual environments, which are isolated spaces where packages can be installed without interfering with each other. This makes it an ideal tool for data scientists who need to work with multiple projects and dependencies. Overall, Anaconda Navigator is a powerful and user-friendly tool that makes managing packages and environments in data science much easier and more efficient.

WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR?

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QTConsole
- Spyder
- VSCode
- Glue viz
- Orange 3 App
- Rodeo
- RStudio
- Advanced conda users can also build your own Navigator applications

PYTHON OVERVIEW:

Python is a High-level Language: Python is a high-level language, which means it abstracts away many of the low-level details of programming such as memory management and garbage collection. This makes it easy to learn and use for beginners and experts alike.

Python has a Large Standard Library: Python's standard library is large and comprehensive, providing functionality for many common programming tasks such as connecting to web servers, reading and writing files, and working with data.

Python is Cross-platform: Python is available for many different operating systems such as Windows, MacOS, and Linux, which means that code written in Python can run on multiple platforms without modification.

Python is Dynamic: Python is a dynamically-typed language, which means that you don't need to specify the data type of a variable when you declare it. The interpreter will automatically determine the data type based on the value assigned to it.

Python is Used in Many Applications: Python is used in a wide variety of applications including web development, scientific computing, data analysis, artificial intelligence, and machine learning. Some popular Python libraries and frameworks include Django, Flask, NumPy, and TensorFlow.

HISTORY OF PYTHON:

Python's first release was in 1991. It quickly gained popularity in the scientific and research community, particularly in the areas of data analysis and visualization. In the 2000s, it started to gain traction in the web development and software development industries. Today, Python is widely used in various fields such as finance, healthcare, and education, and is considered one of the most popular programming languages in the world. Python's simplicity and readability have made it a popular choice for beginners, while its vast collection of libraries and frameworks make it a powerful tool for experienced programmers. The latest version of Python continues to evolve and be updated regularly.

PYTHON FEATURES:

Python's features include:

• It supports modules and packages, allowing for modular and reusable code.

• It has a built-in support for exception handling.

• It has a rich set of libraries and modules for various tasks such as data scraping, data analysis, and machine learning.

• It has a large and active community, which means that there is a vast amount of resources, tutorials, and libraries available.

• It is used by some of the largest companies in the world, such as Google, NASA, and Spotify, which attests to its robustness and scalability.

• It has a versatile application in various domains like Web development, Data Science, AI, Machine Learning, and many more.

• It has a straightforward syntax and easy-to-learn structure making it an ideal language for beginners.

PYTHON ENVIRONMENT:

Python is available on a wide variety of platforms including Linux and Mac OS X.

Python’s standard library

- Pandas
- Numpy
- Sklearn
- seaborn
- matplotlib
- Importing Datasets

PANDAS:

Pandas is quite a game changer when it comes to analyzing data with Python and it is one of the most preferred and widely used tools in data munging/wrangling if not THE most used one. Pandas is an open source

What’s cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example. People who are familiar with R would see similarities to R too). This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

Installation and Getting Started:

In order to “get” Pandas you would need to install it. You would also need to have Python 2.7 and above as a pre-requirement for installation. It is also dependent on other libraries (like NumPy) and has optional dependancies (like Matplotlib for plotting). Therefore, I think that the easiest way to get Pandas set up is to install it through a package like the Anaconda distribution , “a cross platform distribution for data analysis and scientific computing.”

In order to use Pandas in your Python IDE (Integrated Development Environment) like Jupyter Notebook or Spyder (both of them come with Anaconda by default), you need to import the Pandas library first. Importing a library means loading it into the memory and then it’s there for you to work with. In order to import Pandas all you have to do is run the following code:

- import pandas as pd
- import numpy as np

Usually you would add the second part (‘as pd’) so you can access Pandas with ‘pd.command’ instead of needing to write ‘pandas.command’ every time you need to use it. Also, you would import numpy as well, because it is very useful library for scientific computing with Python. Now Pandas is ready for use! Remember, you would need to do it every time you start a new Jupyter Notebook, Spyder file etc.

Working with Pandas:

Loading and Saving Data with Pandas

Pandas is a powerful tool for data analysis and manipulation in Python. It allows you to easily load and work with data in various formats, including CSV, TSV, Excel, and JSON. The process of loading data into a Pandas dataframe can be done in several ways, including converting a Python list, dictionary, or Numpy array, reading a local file, or reading a remote file or database.

To load a local file, you can use the pd.read_filetype() command, where filetype is the specific file format you are working with (e.g. pd.read_csv() for a CSV file). This command can also take various arguments to customize how the file is loaded. To convert a Python object to a Pandas dataframe, you can use the pd.DataFrame() command, where you specify the object you want to convert inside the parentheses.

Once you have loaded your data into a Pandas dataframe, you can use various commands to explore and manipulate the data. The .head() and .tail() commands allow you to view the first or last n rows of the dataframe, respectively. The .shape attribute returns the number of rows and columns in the dataframe, and the .info() command returns information about the dataframe's index, datatype, and memory usage. You can also use the .describe() command to view summary statistics for numerical columns. Other useful commands include .mean(), .corr(), .count(), .max(), .min(), .median(), and .std().

Viewing and Inspecting Data:

Pandas also offers powerful indexing options, such as the ability to select rows and columns by label or by index position. The .loc and .iloc attributes are used for this purpose. For example, df.loc[:, 'column_name'] will select all rows for a specified column, while df.iloc[:, 0] will select all rows for the first column by index position.

Additionally, Pandas provides a variety of methods for filtering, transforming, and manipulating data. These include methods for sorting, aggregating, and grouping data, as well as methods for handling missing data and handling duplicate values.

With these tools and many more, Pandas provides a powerful and flexible framework for data analysis, making it a key component of any data scientist or analyst's toolkit.

Selection of Data

One of the things that is so much easier in Pandas is selecting the data you want in comparison to selecting a value from a list or a dictionary. You can select a column (df[col]) and return column with label col as Series or a few columns (df[[col1, col2]]) and returns columns as a new DataFrame. You can select by position (s.iloc[0]), or by index (s.loc['index_one']) . In order to select the first row you can use df.iloc[0,:] and in order to select the first element of the first column you would run df.iloc[0,0] . These can also be used in different combinations, so I hope it gives you an idea of the different selection and indexing you can perform in Pandas.

Filter, Sort and Groupby

You can use different conditions to filter columns. For example, df[df[year] > 1984] would give you only the column year is greater than 1984. You can use & (and) or | (or) to add different conditions to your filtering. This is also called boolean filtering.

It is possible to sort values in a certain column in an ascending order using df.sort_values(col1) ; and also in a descending order using df.sort_values(col2,ascending=False). Furthermore, it’s possible to sort values by col1 in ascending order then col2 in descending order by using df.sort_values([col1,col2],ascending=[True,False]).

The last command in this section is groupby. It involves splitting the data into groups based on some criteria, applying a function to each group independently and combining the results into a data structure. df.groupby(col) returns a groupby object for values from one column while df.groupby([col1,col2]) returns a groupby object for values from multiple columns.

DATA CLEANING:

Data cleaning is a crucial step in data analysis, as it ensures that the data is accurate and ready for analysis. In Pandas, there are several methods that can be used to clean data. One of the first things to check for is missing values, which can be done by running pd.isnull() to return a boolean array of missing values. To get a count of missing values, you can run pd.isnull().sum(). To remove missing values, you can use the df.dropna() method to drop the rows or df.dropna(axis=1) to drop the columns. Another approach is to fill missing values with a specific value or the mean of the data using the df.fillna() method.

It's also common to replace specific values with new values. This can be done using the s.replace() method, which can take a single value or a list of values to be replaced. Additionally, you can rename columns by using the df.rename() method and change the index of the data frame using df.set_index(). Overall, data cleaning is an important step in data analysis, as it ensures that the data is accurate, complete and ready for analysis.

Join/Combine

Pandas offers several commands for joining or combining data frames or rows/columns. Some of the most commonly used commands include:

- df1.append(df2) - This command adds the rows in df1 to the end of df2, and the columns should be identical.
- df.concat([df1, df2], axis=1) - This command adds the columns in df1 to the end of df2, and the rows should be identical.
- df1.join(df2, on=col1, how='inner') - This command joins the columns in df1 with the columns in df2, where the rows for the specified column (col1) have identical values. The 'how' parameter can be set to 'left', 'right', 'outer', or 'inner' to specify the type of join to perform.

These commands allow you to easily combine different data frames or rows/columns from multiple sources, making it easy to work with and analyze large sets of data.

NUMPY:

Importing NumPy

Numpy is a powerful library for array processing that includes a wide range of high-level mathematical functions. These functions are organized into categories such as Linear Algebra, Trigonometry, Statistics, and Matrix manipulation. To use NumPy, you first need to install it. The main object in NumPy is a homogeneous multidimensional array, which is more versatile than python's standard array class, which only supports one-dimensional arrays. NumPy arrays have multiple dimensions, or axes. For example, a two-dimensional array has rows and columns as its axes. These dimensions can also be referred to as the rank of the array or matrix.

SKLEARN

In python, scikit-learn library has a pre-built functionality under sklearn. Pre processing.

Next thing is to do feature extraction Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally our models are trained using Classifier algorithm.. We use nltk . classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered . The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre processed data. The chosen classifiers were Decision tree , Support Vector Machines and Random forest. These algorithms are very popular in text classification tasks.

SEABORN:

Data Visualization in Python:

Data visualization is the process of understanding data by placing it in a visual context, allowing patterns, trends, and correlations to be easily identified. Python offers a variety of powerful graphing libraries that come packed with many different features. Whether you want to create interactive, live, or highly customized plots, Python has an excellent library for you. Some popular plotting libraries include:

- Matplotlib: a low-level library that provides a lot of flexibility.
- Pandas Visualization: an easy-to-use interface built on top of Matplotlib.
- Seaborn: a high-level interface with great default styles.
- ggplot: based on R's ggplot2 and uses the Grammar of Graphics.
- Plotly: can create interactive plots. In this article, we will learn how to create basic plots using Matplotlib, Pandas visualization, and Seaborn, as well as how to use some specific features of each library. The focus of this article will be on the syntax, not on interpreting the graphs.

MATPLOTLIB:

Matplotlib is the most popular python plotting library. It is a low level library with a Matlab like interface which offers lots of freedom at the cost of having to write more code.

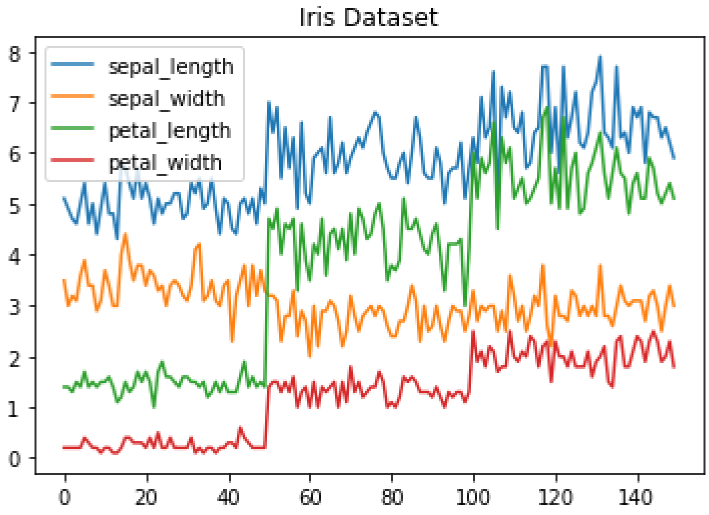
- To install Matplotlib pip and conda can be used.
- pip install matplotlib
- conda install matplotlib

Matplotlib is specifically good for creating basic graphs like line charts, bar charts, histograms and many more. It can be imported by typing:

import matplotlib.pyplot as plt

LINE CHART

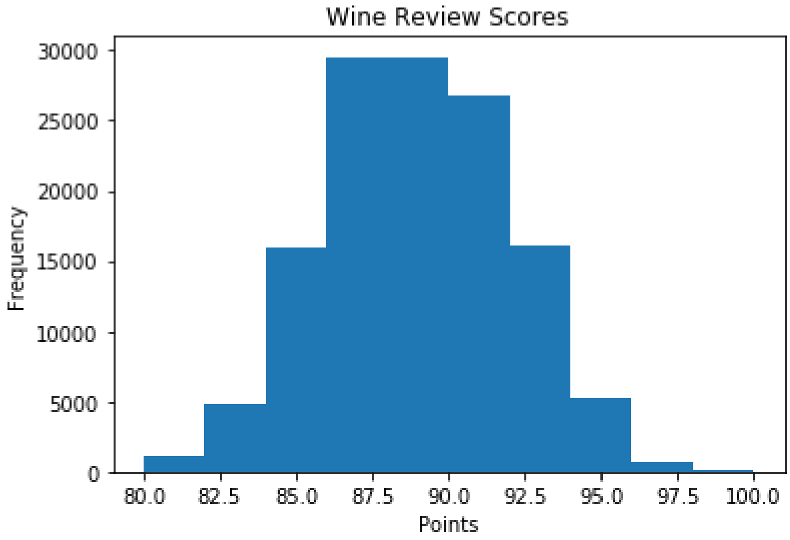
In Matplotlib we can create a line chart by calling the plot method. We can also plot multiple columns in one graph, by looping through the columns we want, and plotting each column on the same axis.



Line Chart

HISTOGRAM:

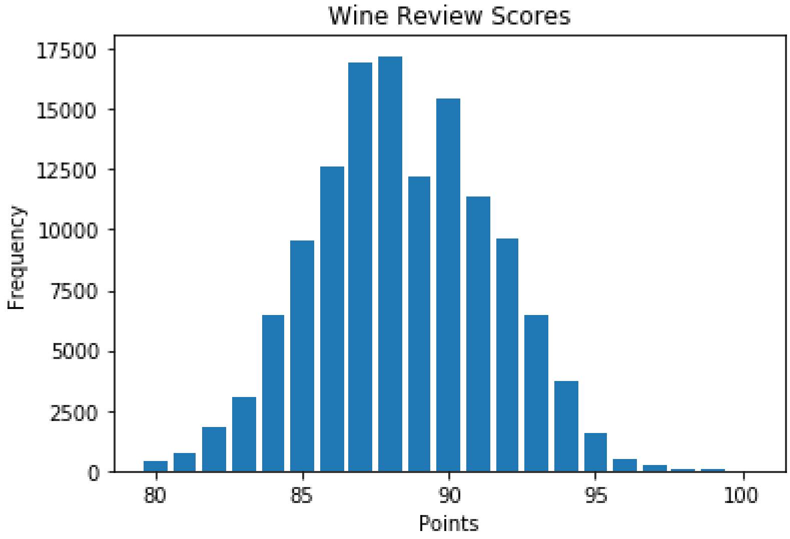
In Matplotlib we can create a Histogram using the hist method. If we pass it categorical data like the points column from the wine-review dataset it will automatically calculate how often each class occurs.



Histogram

BAR CHART:

A bar-chart can be created using the bar method. The bar-chart isn't automatically calculating the frequency of a category so we are going to use pandas value_counts function to do this. The bar-chart is useful for categorical data that doesn't have a lot of different categories (less than 30) because else it can get quite messy.



Bar-Chart

PANDAS VISUALIZATION:

Pandas is a open source high-performance, easy-to-use library providing data structures, such as dataframes, and data analysis tools like the visualization tools we will use in this article.

Pandas Visualization makes it really easy to create plots out of a pandas dataframe and series. It also has a higher level API than Matplotlib and therefore we need less code for the same results.

- Pandas can be installed using either pip or conda.
- pip install pandas
- conda install pandas

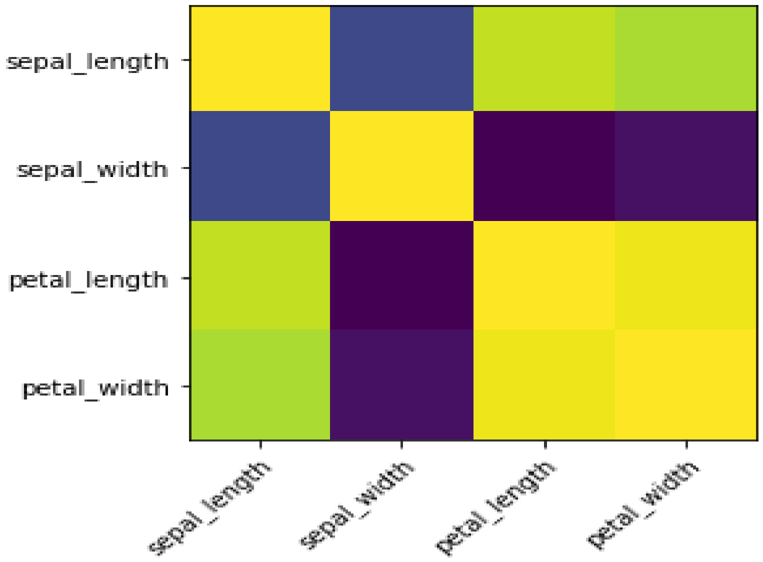
HEATMAP:

A Heatmap is a graphical representation of data where the individual values contained in a matrix are represented as colors. Heatmaps are perfect for exploring the correlation of features in a dataset.

To get the correlation of the features inside a dataset we can call <dataset>.corr(), which is a Pandas dataframe method. This will give us the correlation matrix.

We can now use either Matplotlib or Seaborn to create the heatmap.

MATPLOTLIB:



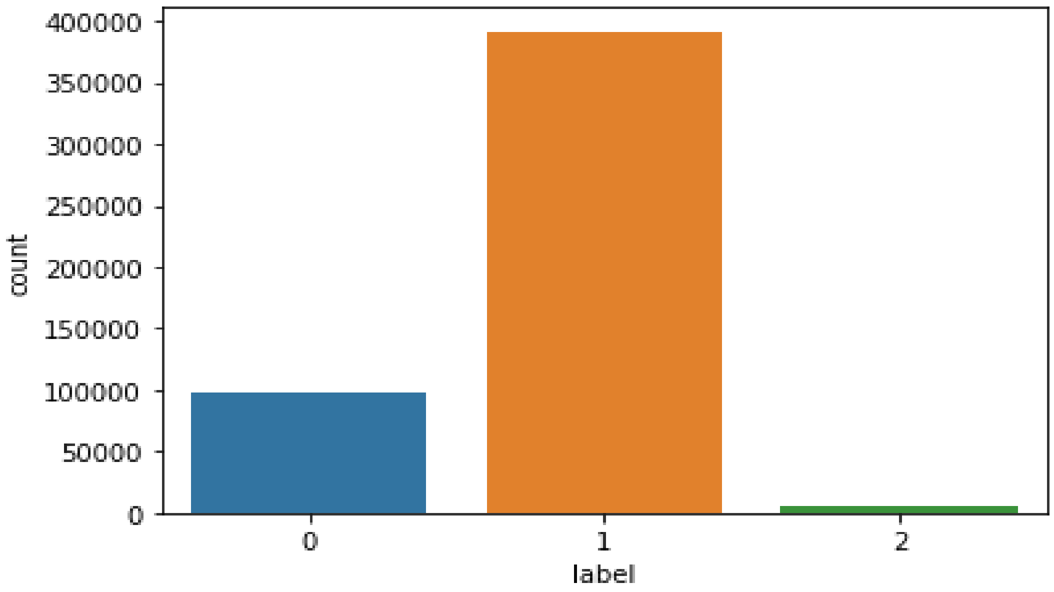
Heatmap without annotations

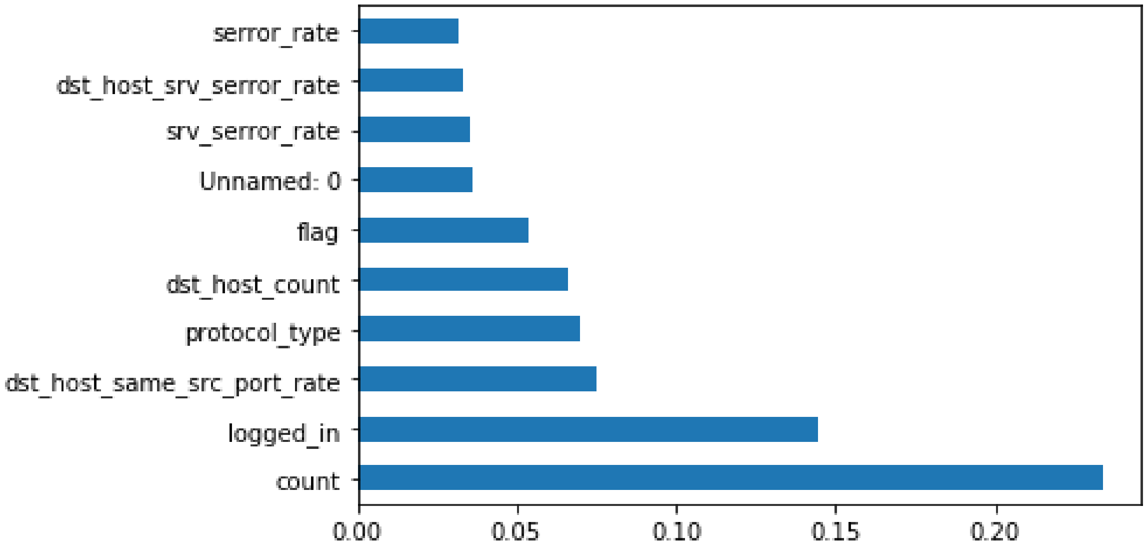
Data visualization is the discipline of trying to understand data by placing it in a visual context so that patterns, trends, and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries packed with many different features. In this article, we looked at Matplotlib, Pandas visualization, and Seaborn.

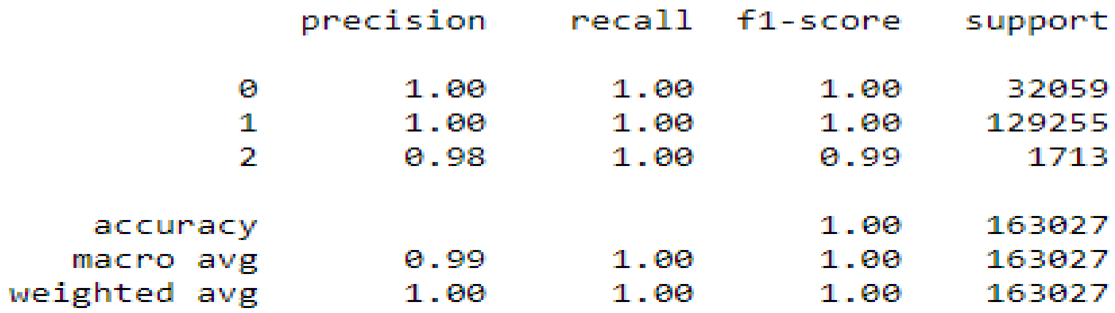
OUTPUT RESULTS:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c4ade44d08>
```





RANDOM FOREST RESULT:

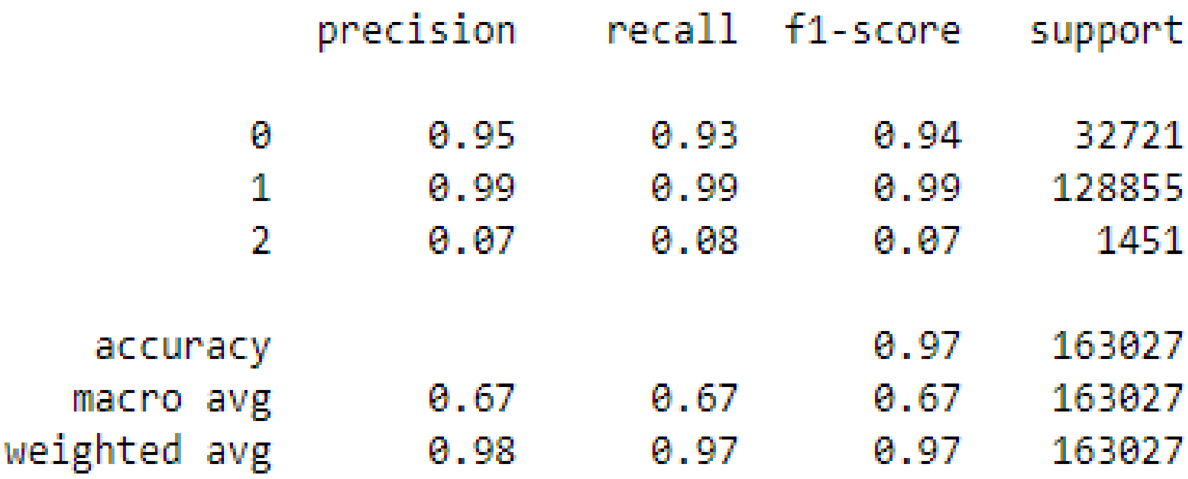


```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     32059
           1       1.00      1.00      1.00    129255
           2       0.98      1.00      0.99      1713

    accuracy                           1.00    163027
   macro avg       0.99      1.00      1.00    163027
weighted avg       1.00      1.00      1.00    163027
```

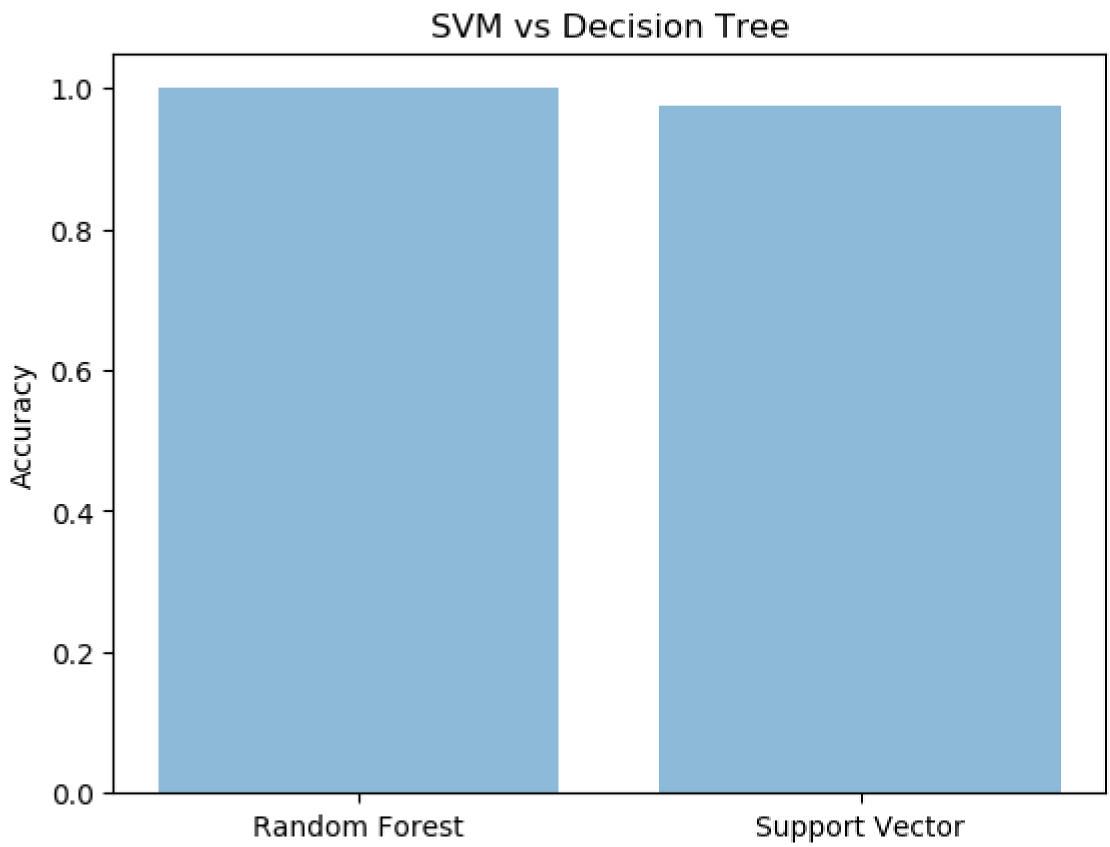
SVM RESULT:



```
              precision    recall  f1-score   support

           0       0.95      0.93      0.94     32721
           1       0.99      0.99      0.99    128855
           2       0.07      0.08      0.07      1451

    accuracy                           0.97    163027
   macro avg       0.67      0.67      0.67    163027
weighted avg       0.98      0.97      0.97    163027
```



References:

Machine Learning for Intrusion Detection: A Review, Wei Lu, Dongmei Zhang, and Xinyu Xing

Intrusion Detection in Computer Networks using Machine Learning Techniques, A. K. Verma and Jyoti Bajpai

A Survey of Machine Learning Techniques for Network Intrusion Detection, S. Singh, K. Kaur, and A. Singh

Advanced Persistent Threats: Understanding the Danger, http:// www.sans.org /reading-room/whitepapers/threats/advanced-persistent-threats-understanding-danger-33229