A SHORT-TERM INTERNSHIP REPORT ON

ARTIFICIAL INTELLIGENCE &

MACHINE LEARNING

By-

NEELI PREM KUMAR

KONDEPUDI VINAY KUMAR

B.D.V.S. SIVA PRASAD

KEELA VAMSIDHAR

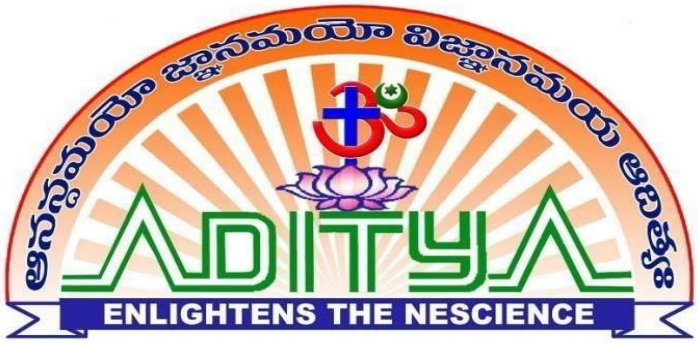
III BCA [2022-2025]

Under the Esteemed

Guidance of Mr. G.V.S.S

PRASANTH SIR

(Tutor of Artificial Intelligence & Machine Learning)



ADITYA DEGREE COLLEGE

[COED], GAJUWAKA.

(Affiliated to Andhra University)

Gajuwaka-530026, Visakhapatnam District,

ANDHRA PRADESH.

ADITYA DEGREE COLLEGE

DECLARATION BY THE STUDENT

We hereby declare that the work described in this short-term Internship, entitled “Artificial Intelligence & Machine Leaning”, Which is being submitted by us in partial fulfilment of the requirements for the award of degree of Bachelor of Computer Applications from the department of Bachelor of Computer Science to Aditya Degree College, Gajuwaka under the guidance of Mr. G.V.S.S Prasanth Sir tutor of Artificial Intelligence & Machine Learning in Aditya Degree College, Gajuwaka.

Place: Gajuwaka Date:

31-07-2024

(Neeli Prem Kumar

Kondepudi Vinay Kumar

B.D.V.S. Siva Prasad

Keela Vamsidhar)

ADITYA DEGREE COLLEGE

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the Short-Term Internship entitled, "**Artificial Intelligence and Machine Learning**", that is being submitted by Neeli Prem Kumar (1221206060), Kondepudi Vinay Kumar (1221206060), B.D.V.S. Siva Prasad(1221206060), Keela Vamsidhar (122120606052) of **III BCA** which is being submitted to us in partial fulfilment of the requirements for the award of degree of **Bachelor of Computer Applications** from the department of Bachelor of Computer Science to Aditya Degree College, bonified work carried out by them under my guidance and supervision.

(Mr. G. V. S. S Prasanth Sir)

ACKNOWLEDGEMENT

No endeavour is completed without the valuable support of others.

We would like to take this opportunity to extend our sincere gratitude to all those who have contributed to the successful completion of this Short

Term Internship Project Report.

We express our deep sense of gratitude to **Mr. Satya Prakash Sir**, **Principal**, for his efforts and for giving us permission for carrying out this Short Term Internship.

We feel deeply honoured in expressing our sincere thanks to **Mr, G.V.S.S Prasanth Sir** tutor of **ULearn** for making the resources available at right time and providing valuable insights leading to the successful completion of Short-Term Internship Project Report.

Finally, we thank all the faculty members of our department who contributed their valuable suggestion in completion of Short-Term Internship Report and We also put our sincere thanks to our parents who stood with us during the whole Short-Term Internship.

(Neeli Prem Kumar, Kondepudi Vinay Kumar,

B.D.V.S. Siva Prasad, Keela Vamsidhar)

CONTENTS

- Parameter sharing and typing
- Ensemble methods
- Bayes theorem
- LSTM- long short-term memory
- Restricted Boltzmann Machine
- RNN- Recurrent Neural Network
- Auto encoders and types
- VGG Net and architecture
- Google Net and architecture
- Data types in Python
- Arithmetic operations in python
- Declaration of comments and variables
- Reserved words in python
- Control statements in python
- Programs

- **Problem statement & Explanation**
- **Source and Outputs**
- **Conclusion**

INTRODUCTION

Churn is the measure of how many customers stop using a product. This can be measured based on actual usage or failure to renew (when the product is sold using a subscription model). Often evaluated for a specific period of time, there can be a monthly, quarterly, or annual churn rate.

When new customers start buying and/or using a product in a bank, each new user contributes to the bank product's growth rate. Certainly, some of those customers in due course will stop their utilization or end their subscription; this could be because they switched to a competitor, no longer need the bank services, they're unhappy with their user experience, or they can no longer afford the cost. The customers that stop using the bank products are the "churn" for a given period. Which can be can be a monthly, quarterly, or annual churn rate at the bank.

As we know, it is much more expensive to sign in a new client than keeping an existing one and the fact that more profits are produced through long-term costumers. Therefore, customer retention increases profitability. Many competitive companies have noticed that a key approach for survival within the industry is to retain existing customers. This leads to the importance of churn management in organizations such as a bank.

We began this analysis with goals to discover key insights from the bank customers database and study the customers' demographics such as customer (gender, age, and location). Also, we incline to understand the company's product and customer's financial history such as customer (credit score, estimated salary, balance, tenure, credit card possession, etc.). Lastly, how variable such as customers demographics and financial history affects the customers churn rate.

We will be performing analysis and developing a **Prediction model for bank customer churn.**

LEARNING OUTCOME OF

SHORT-TERM INTERNSHIP

Introduction to AI & ML: -

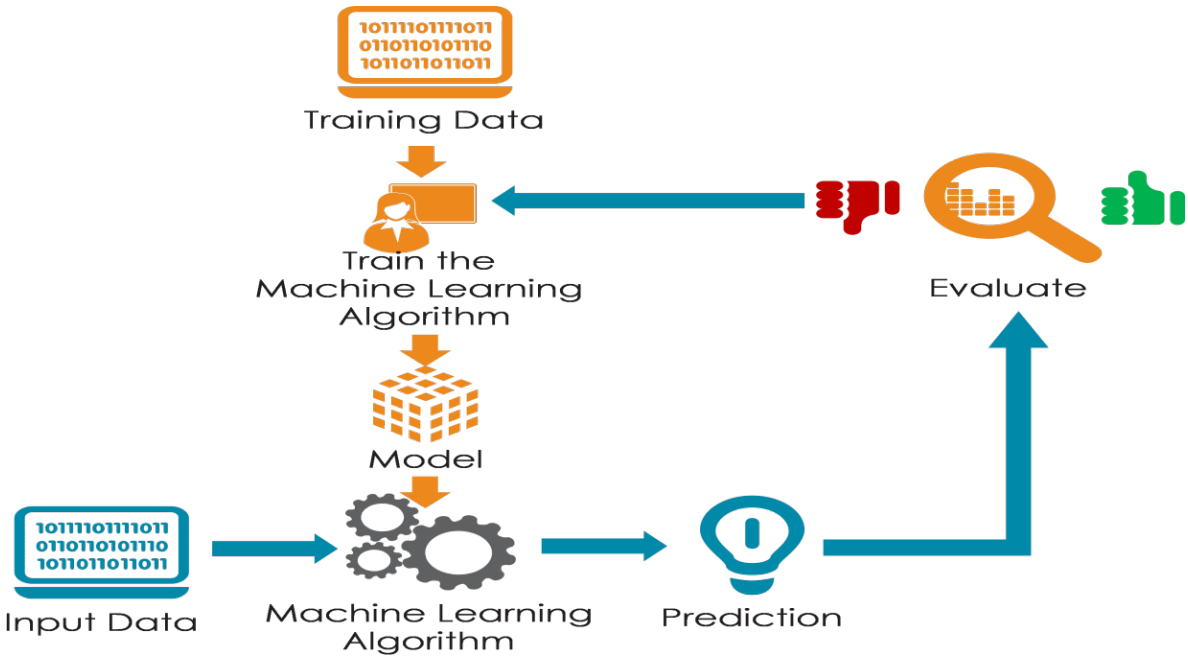
AI is a branch of computer science. It stands for artificial intelligence. It is the simulation of human intelligence which is processed by machines.

There are two subsets in AI. They are: - 1. Machine Learning 2. Deep Learning

Machine Learning: -

Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data and thus perform tasks without explicit instructions. Recently, artificial neural networks have been able to surpass many previous approaches in performance.

ML finds application in many fields, including natural language processing, computer vision, speech recognition, email filtering, agriculture, and medicine. When applied to business problems, it is known under the name predictive analytics. Although not all machine learning is statistically based, computational statistics is an important source of the field's methods.

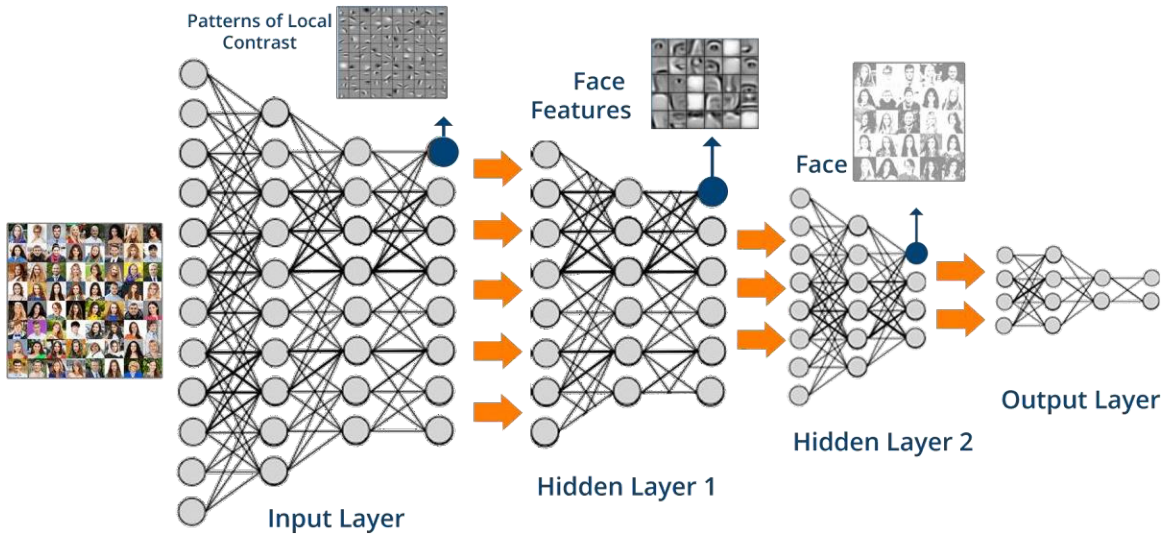


Deep Learning: -

Deep learning is the subset of machine learning methods based on neural networks with representation learning. The adjective "deep" refers to the use of multiple layers in the network. Methods used can be either supervised, semi-supervised or unsupervised.

Deep-learning architectures such as deep neural networks, deep belief networks, recurrent neural networks, convolutional neural networks and transformers have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection and board

game programs, where they have produced results comparable to and in some cases surpassing human expert performance.



Types of ML: -

There are three types of machine learning:

1. supervised learning – it is a labelled data or structured data
2. unsupervised learning – it is un-labelled data or unstructured data
3. reinforcement learning -it uses both structured data and unstructured data

Supervised learning: -

- Supervised learning involves training a machine from labelled data.
- Labelled data consists of examples with the correct answer or classification.
- The machine learns the relationship between inputs and outputs.
- The trained machine can then make predictions on new, unlabelled data.

  Supervised learning is classified into two categories of algorithms: • Regression:

  A regression problem is when the output variable is a real value, such as “dollars” or “weight”.
- Classification: A classification problem is when the output variable is a category, such as “Red” or “blue”, “disease” or “no disease”

Regression: -

Regression is a type of supervised learning that is used to predict continuous values, such as house prices, stock prices, or customer churn. Regression algorithms learn a function that maps from the input features to the output value.

Some common regression algorithms include:

- Linear Regression
- Polynomial Regression
- Support Vector Machine Regression
- Decision Tree Regression
- Random Forest Regression

Classification: -

Classification is a type of supervised learning that is used to predict categorical values, such as whether a customer will churn or not, whether an email is spam or not, or whether a medical image shows a tumour or not. Classification algorithms learn a function that maps from the input features to a probability distribution over the output classes.

Some common classification algorithms include:

- Logistic Regression
- Support Vector Machines
- Decision Trees
- Random Forests
- Naive Baye

Unsupervised Learning: -

- Unsupervised learning allows the model to discover patterns and relationships in unlabelled data.
- Clustering algorithms group similar data points together based on their inherent characteristics.
- Feature extraction captures essential information from the data, enabling the model to make meaningful distinctions.
- Label association assigns categories to the clusters based on the extracted patterns and characteristics.

Unsupervised learning is classified into two categories of algorithms:

- Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour.
- Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Clustering: -

Clustering is a type of unsupervised learning that is used to group similar data points together. Clustering algorithms work by iteratively moving data points closer to their cluster centres and further away from data points in other clusters. Some types of clustering are:

- Hierarchical clustering
- K-means clustering
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis

Association: -

Association rule learning is a type of unsupervised learning that is used to identify patterns in a data. Association rule learning algorithms work by finding relationships between different items in a dataset.

Some common association rule learning algorithms include:

- Apriori Algorithm
- Eclat Algorithm
- FP-Growth Algorithm

Reinforcement Learning: -

Reinforcement Learning (RL) is the science of decision making. It is about learning the optimal behaviour in an environment to obtain maximum reward. In RL, the data is accumulated from machine learning systems that use a trial-and- error method. Data is not part of the input that we would find in supervised or unsupervised machine learning.

Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next.

Reinforcement learning elements are as follows:

- Policy
- Reward function
- Value function
- Model of the environment

Applications of ML: -

Today, companies are using Machine Learning to improve business decisions, increase productivity, detect disease, forecast weather, and do many more things. Some of the most common examples are:

- Image Recognition
- Speech Recognition
- Recommender Systems
- Fraud Detection
- Self-driving Cars
- Medical Diagnosis
- Stock Market Trading
- Virtual Try On

Image Recognition: -

Image recognition made a bloom in Deep Learning. The task which started from classification between cats and dog images has now evolved up to the level of Face Recognition and real-world use cases based on that like employee attendance tracking.

Speech Recognition: -

Speech Recognition based smart systems like Alexa and Siri have certainly come across and used to communicate with them. In the backend, these systems are based basically on Speech Recognition systems. These systems are designed such that they can convert voice instructions into text.

Recommender Systems: -

Approximately everyone trying to provide customized services to its users. This application is possible just because of the recommender systems which can analyse a user's preferences and search history and based on that they can recommend content or services to them.

Fraud detection: -

Due to ML applications only whenever the system detects red flags in a user's activity than a suitable notification be provided to the administrator so, that these cases can be monitored properly for any spam or fraud activities.

Medical diagnosis: - Not even in the field of disease diagnosis in human beings but they work perfectly fine for plant disease-related tasks whether it is to predict the type of disease it is or to detect whether some disease is going to occur in the future.

Deep Learning: -

In the fast-evolving era of artificial intelligence, Deep Learning stands as a cornerstone technology, revolutionizing how machines understand, learn, and interact with complex data.

At its essence, Deep Learning AI mimics the intricate neural networks of the human brain, enabling computers to autonomously discover patterns and make decisions from vast amounts of unstructured data.

This transformative field has propelled breakthroughs across various domains, from computer vision and natural language processing to healthcare diagnostics and

autonomous driving. In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after the other.

ANN, NLP, CC: -

Artificial neural networks: -

Artificial Neural Networks contain artificial neurons which are called units. These units are arranged in a series of layers that together constitute the whole Artificial Neural Network in a system. A layer can have only a dozen units or millions of units as this depends on how the complex neural networks will be required to learn the hidden patterns in the dataset. Commonly, Artificial Neural Network has an input layer, an output layer as well as hidden layers. The input layer receives data from the outside world which the neural network needs to analyse or learn about. Then this data passes through one or multiple hidden layers that transform the input into data that is valuable for the output layer. Finally, the output layer provides an output in the form of a response of the Artificial Neural Networks to input data provided.

Natural language processing (NLP):

In Deep learning applications, second application is NLP. NLP, the Deep learning model can enable machines to understand and generate human language. Some of the main applications of deep learning in NLP include:

- Automatic Text Generation: -Deep learning model can learn the corpus of text and new text like summaries, essays can be automatically generated using these trained models.
- Language translation: -Deep learning models can translate text from one language to another, making it possible to communicate with people from different linguistic backgrounds.
- Sentiment analysis: -Deep learning models can analyse the sentiment of a piece of text, making it possible to determine whether the text is positive, negative, or neutral. This is used in applications such as customer service, social media monitoring, and political analysis.
- Speech recognition: -Deep learning models can recognize and transcribe spoken words, making it possible to perform tasks such as speech-to-text conversion, voice search, and voice-controlled devices.

Congestion Control: -

Congestion Control is a mechanism that controls the entry of data packets into the network, enabling a better use of a shared network infrastructure and avoiding congestive collapse. Congestive-Avoidance Algorithms (CAA) are implemented at the TCP layer as the mechanism to avoid congestive collapse in a network. There are two congestion control algorithm which are as follows:

- Leaky Bucket Algorithm: -

The leaky bucket algorithm discovers its use in the context of network traffic shaping or rate-limiting. This algorithm is used to control the rate at which traffic is sent to the network and shape the burst traffic to a steady traffic stream. The large area of network resources such as bandwidth is not being used effectively.

- Token bucket Algorithm: -

In some applications, when large bursts arrive, the output is allowed to speed up. This calls for a more flexible algorithm, preferably one that never loses information. Therefore, a token bucket algorithm finds its uses in network traffic shaping or rate-limiting. It is a control algorithm that indicates when traffic should be sent. This order comes based on the display of tokens in the bucket.

AI Tools In Our Daily Life: -

In our daily life, we use lot of AI tools. These AI tools help us to work in an effective manner.

Following is some of the AI tools we use on our daily basis: -

Voice assistance: -

Voice assistance is used to communicate with the computer. It recognizes the human language and coverts into machine understandable language. It uses NLP technology i.e. Natural language processing.

Examples are Alexa, Siri, Google Assistance, Bixby etc.

Image recognition: -

It identifies the object in the image. It recognizes the object by using artificial intelligence. It also uses OCR technology i.e. optical character recognition. OCR is used to detect the spam mails by using POS tagging.

Examples are Google lens, X-rays, spam detection etc.

CHATGPT: -

CHATGPT is a generative AI tool which is used to get information related to the question. It is like a chat-box.

Some of other tools like these are SNAPAI, Copilot, Gemini etc.

Web browsers: - web browsers are like search engines that gives information relevant to our query. These are predictive AI too.

Examples are Google, Microsoft edge, Firefox, Yahoo etc.

Recommendation systems: -

Recommendation systems refer to that they give recommendations relevant to our searches. They analyse our search history, watch history and likes on the post and gives the recommendations based on it.

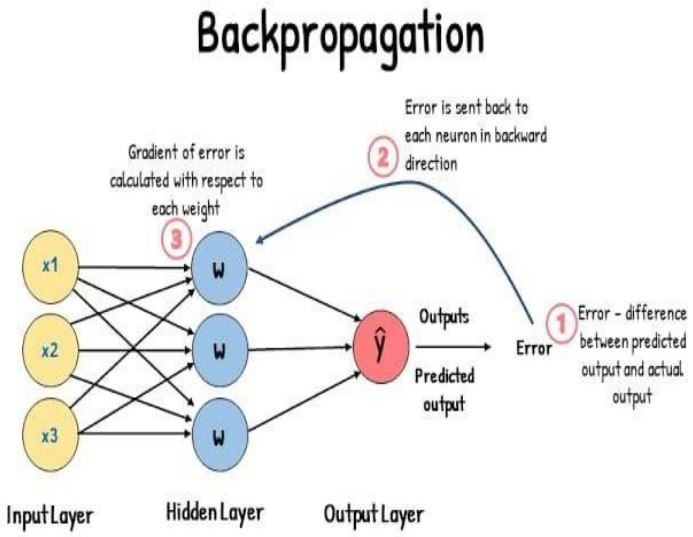
Examples are YouTube, Instagram, Aha, Flipkart and all other apps.

Back Propagation: -

The reverse process of feed forward neural network is called back propagation. It rectifies the error in hidden layers and form backward direction. The flow chart as follows: -

Predictive value errors (input layer) ★backward direction ★output layer to hidden layer ★errors correction ★output layer ★prediction value

Calculation of weights: -



following are the terms to keep in mind while calculating weights: -

•Weights should be in decimal value.

•Sum of neurons or predictive values is called bias.

•From the adjacent diagram, the terms are: -

•i1 and i2 are input values

•h1 and h2 are hidden layers

•o1 and o2 are output values

•b1 and b2 are bias values

•w1, w2, ……. are weights

Now for NETh1= w1*i1 + w2*i2 + b1*1

= 0.15*0.05 + 0.20*0.10 + 0.35*1 = 0.3775

Now for NETh2= w3*i1 + w4*i2 + b2*1

= 0.25*0.05 + 0.30*0.10 + 0.60*1= 0.645

Difference between Neural and Deep Neural Networks: -

The differences between the neural networks and deep learning neural networks are tabulate as follows: -

s.no	Differences in	Neural Networks	Deep Learning Neural Networks
1.	Definition	A neural network is a model of neurons inspired by the human brain. It is made up of many neurons that at inter-connected with each other.	Deep learning neural networks are distinguished from neural networks on the basis of their depth or number of hidden layers.
2.	Architecture	Feed Forward Neural Networks Recurrent Neural Networks Symmetrically Connected Neural Networks	Recursive Neural Networks Unsupervised Pre-trained Networks Convolutional Neural Networks
3.	Structure	Neurons Connection and weights Propagation function Learning rate	Motherboards PSU RAM Processors
4.	Performance	It gives low performance compared to Deep Learning Networks.	It gives high performance compared to neural networks.

5.	Task Interpretation	Your task is poorly interpreted by a neural network.	The deep learning network more effectively.

Difference between CHATGPT & GOOGLE:

The differences between CHATGPT & GOOGLE are tabulated as follow: -

s.no	CHATGPT	GOOGLE
1.	CHATGPT is an AI powered tool.	GOOGLE is a search engine.
2.	It works like a chat box between the user and server.	It gives information by showing different websites.
3.	It gives the information based on the question entered.	It shows the relevant information in different sites.
4.	Data may not be accurate	Gives most accurate data.
5.	It gives the answer based on the information it trained on.	Gives the answer based on the searches and reviews.
6.	It focused on generating humanlike texts.	It can be used for variety of tasks like voices and image recognition.
7.	It provides an answer based on the personal views and subjective views found in the data.	It provides information based on the Articles opinions of experts and activists.
8.	It is an Artificial intelligence model.	It is a worldwide search engine.
9.	Gives the information from its data source.	Gives the information that is already on the internet
10.	It is developed by OpenAI.	It is developed by Google Inc.

POS Tagging: -

Parts of Speech tagging is a linguistic activity in Natural Language Processing

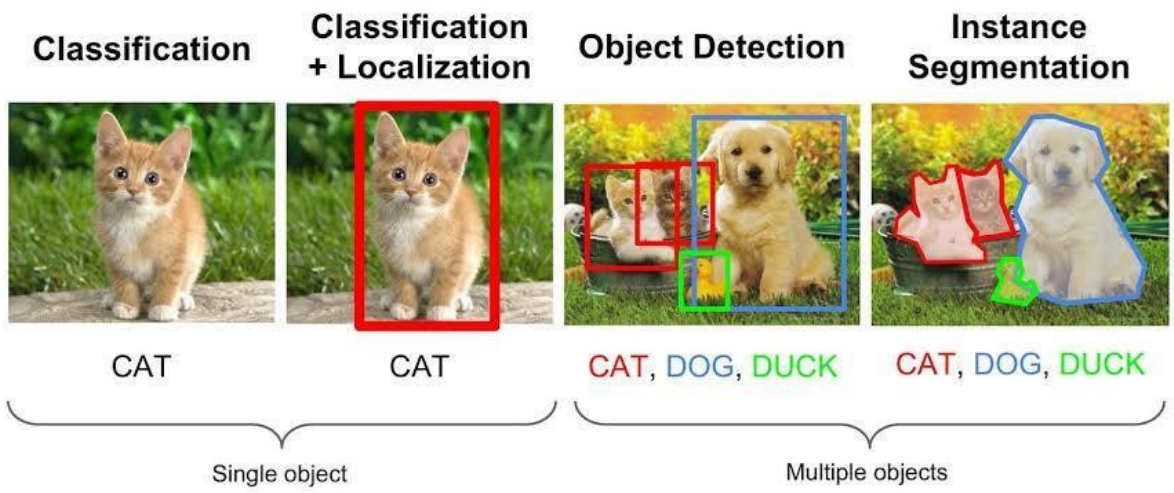
(NLP) wherein each word in a document is given a particular part of speech (adverb, adjective, verb, etc.) or grammatical category. Through the addition of a layer of syntactic and semantic information to the words, this procedure makes it easier to comprehend the sentence's structure and meaning.

In NLP applications, POS tagging is useful for machine translation, named entity recognition, and information extraction, among other things. It also works well for clearing out ambiguity in terms with numerous meanings and revealing a sentence's grammatical structure.

POS tagging stands for Part-of-Speech tagging. It's like giving each word in a sentence a label that shows what part of speech it is. For example, if you have a sentence like "The cat is sleeping," POS tagging would label "The" as a determiner, "cat" as a noun, "is" as a verb, and "sleeping" as a verb too. It helps computers understand the meaning of words in a sentence.

Object Detection: -

- Object detection has been a important topic in advancement of computer vision systems.
- With the advent of deep learning techniques, the cure for object detection has increased drastically.
- The project aims to incorporate state-of-the-art technique for object detection with the goal of achieving high accuracy with a real-time performance.
- A major challenge in many of the object detection systems is the dependency on other computer vision techniques for helping the deep learning-based approach, which leads to slow and non-optimal performance.
- In this project, we use a completely deep learning approach to solve the problem of object detection in an end-to-end fashion.
- The network is trained on the most challenging publicly available dataset (PASCAL VOC), on which an object detection challenge is conducted annually.
- The resulting system is fast and accurate, thus along those applications which require object detection.
- Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results.
- When humans look at images or video, we can recognize and locate objects of interest within a matter of moments. The goal of object detection is to replicate this intelligence using a computer.



CNN Algorithm: -

A Convolutional Neural Network (CNN), also known as ConvNet, is a specialized type of deep learning algorithm mainly designed for tasks that necessitate object recognition, including image classification, detection, and segmentation. CNNs are employed in a variety of practical scenarios, such as autonomous vehicles, security camera systems, and others.
The convolutional neural network is made of four main parts. They help the CNNs mimic how the human brain operates to recognize patterns and features in images:

- Convolutional layers
- Rectified Linear Unit (ReLU for short)
- Pooling layers
- Fully connected layers

Deep Fake & Deep Dream: -

"Deep fake" and "Deep Dream" are two distinct concepts within the realm of artificial intelligence and deep learning.

1. Deep Fake:

• Description:

Deep fakes refer to synthetic media where a person in an existing image or video is replaced with someone else's likeness. They leverage deep learning techniques, particularly generative adversarial networks (GANs), to create realistic but fake videos, images, or audio.

• Use Cases:

Often used in entertainment, such as in movies to create digital characters, but also notorious for creating fake news, disinformation, and non- consensual explicit content.

• Concerns:

Ethical and legal issues, potential for misuse in spreading misinformation or committing fraud.

2. Deep Dream:

• Description:

Deep Dream is a computer vision program created by Google which uses a convolutional neural network (CNN) to find and enhance patterns in images through algorithmic pareidolia, creating a dream-like, surrealistic effect.

• Use Cases:

Primarily used in artistic contexts to generate visually striking and abstract imagery, turning ordinary photos into dream-like versions.

• Concerns:

Mainly artistic, with less direct ethical implications compared to deep fakes.

Both technologies demonstrate the capabilities of deep learning but are applied in very different contexts with varying implications.

GAN Model & Architecture: -

GAN model: -

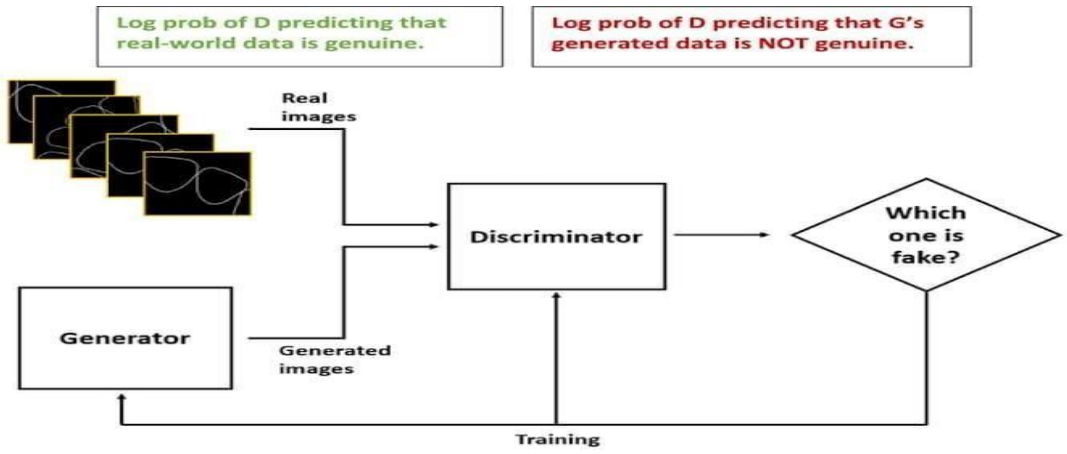
- A generative adversarial network (GAN) is a deep learning architecture.020
- It trains two neural networks to compete against each other to generate more authentic new data from a given training dataset.020
- For instance, you can generate new images from an existing image database or original music from a database of songs.020
- A GAN is called adversarial because it trains two different networks and pits them against each other.020
- One network generates new data by taking an input data sample and modifying it as much as possible.020

- The other network tries to predict whether the generated data output belongs in the original dataset.020

- In other words, the predicting network determines whether the generated data is fake or real.020

- The system generates newer, improved versions of fake data values until the predicting network can no longer distinguish fake from original.020

GAN architecture:



A Generative Adversarial Network (GAN) is composed of two primary parts, which are the Generator and the Discriminator.

Generator Model-

A key element responsible for creating fresh, accurate data in a Generative Adversarial Network (GAN) is the generator model. The generator takes random noise as input and converts it into complex data samples, such text or images. It is commonly depicted as a deep neural network. The generator's ability to generate high-quality, varied samples that can fool the discriminator is what makes it successful.

Generator Loss:

The objective of the generator in a GAN is to produce synthetic samples that are realistic enough to fool the discriminator. The generator achieves this by minimizing its loss function. When the discriminator is highly likely to classify the generated samples as real.

Discriminator Model-

An artificial neural network called a discriminator model is used in GANs to differentiate between generated and actual input. By evaluating input samples and allocating probability of authenticity, the discriminator functions as a binary classifier. Over time, the discriminator learns to differentiate between genuine data from the dataset and artificial samples created by the generator. This allows it to progressively hone its parameters and increase its level of proficiency.

Discriminator Loss:

The discriminator reduces the negative log likelihood of correctly classifying both produced and real samples. This loss incentivizes the discriminator to accurately categorize generated samples as fake and real samples.

Data Augmentation: -

Data augmentation is a technique used in machine learning and deep learning to increase the diversity of a training dataset without collecting new data. By applying various transformations to the existing data, data augmentation helps improve the performance and generalization of machine learning models. Here are some common methods and their benefits:

Common Data Augmentation Techniques- 1.
Image Data Augmentation:

- Flipping: Horizontally or vertically flipping images.
- Rotation: Rotating images by a certain angle.
- Scaling: Zooming in or out of images.
- Translation: Shifting images horizontally or vertically.
- Shearing: Applying a shear transformation to images.
- Cropping: Randomly cropping parts of the image.
- Colour Jittering: Changing the brightness, contrast, saturation, and hue. - Adding Noise: Introducing random noise to images.

2. Text Data Augmentation:

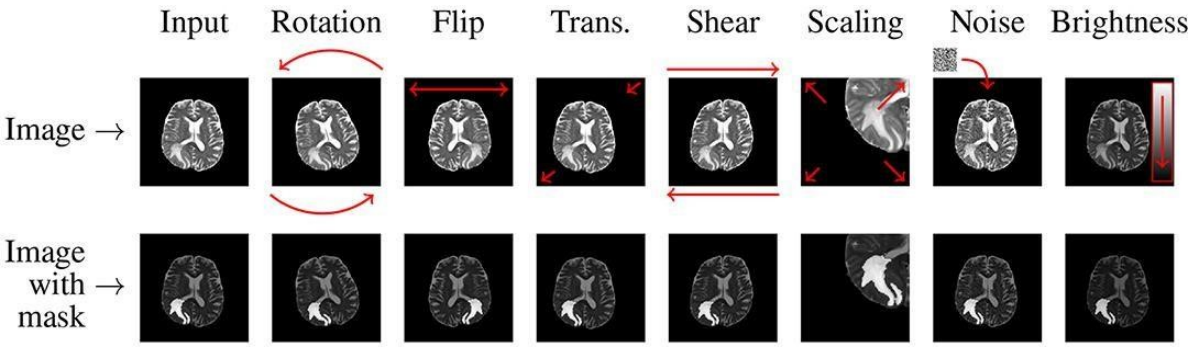
- Synonym Replacement: Replacing words with their synonyms.
- Random Insertion: Inserting random words at random positions.
- Random Deletion: Deleting words randomly from sentences.
- Back Translation: Translating text to another language and back to the original language.

3. Time Series Data Augmentation:

- Window Slicing: Taking random slices of the time series.
- Time Warping: Distorting the time intervals of the data.
- Adding Noise: Introducing random noise to the time series data. - Scaling: Changing the scale of the data.

Benefits of Data Augmentation-

- Increased Dataset Size: Augmenting the data increases the size of the dataset, which can help prevent overfitting, especially when the original dataset is small.
- Improved Model Generalization: Models trained with augmented data tend to generalize better to unseen data, improving their robustness and accuracy.
- Reduced Overfitting: By exposing the model to more varied data, data augmentation helps reduce overfitting to the training set.



Parameter Sharing & Typing: -

It is a convolutional neural network model which is used to share the weights equally in neural networks.

- It is a deep learning application.
- Parameter sharing is the method of sharing weights by all neurons in a particular feature map.
- It helps to reduce the number of parameters in the whole system. • Parameter sharing is used in all convolution layers in the network.
- It reduces the training time.
- The idea behind parameter sharing is the essence of forcing the parameters to be similar.
- In parameter typing, two models performing the same classification task but with somewhat different input distributions.
- Parameter typing refers to the practice of constraining different parts of a model to share the same parameter values.
- This can be useful in cases where we want to encourage certain properties of model, such as symmetry or sparsity.

Ensemble Methods: -

Ensemble methods are techniques that aims to improve the results in models by combining multiple models instead of single model. These methods help to increase the accuracy of the results. Ensemble methods are ideal for regression and classification where they reduce bis and variance to accuracy of models. The most popular ensemble methods are as follows: -

Bagging: -

★Bagging is the short form for bootstrap aggregating. It is mainly applied in classification and regression.

★It increases the accuracy of models through decision trees, which reduces variance to a large extent.

★It classified into two types i.e. bootstrapping and aggregation.

★Bootstrapping is a sampling technique where samples are derived from the whole set using the replacement procedure.

★Aggregation in bagging is done to incorporate all possible outcomes of the prediction and randomize the outcome.

★Without aggregation, predictions will not be accurate because all outcomes are not put into consideration.

Boosting: -

★Boosting is an ensemble method that learns from previous predictor mistakes to make better predictions in the future.

★This technique combines several weak base learners to form one strong learner, thus improving the predictability of models.

★Boosting takes many forms, including gradient boosting, Adaptive boosting, and XG Boost.

Stacking: -

★Stacking is often referred to as stacked generalization.

★This method works by allowing a training algorithm to ensemble several other similar learning algorithm predictions.

★It can also be used to measure the error rate involved during bagging.

★Combination of boosting and stacking is called Boo stacking.

Bayes Theorem: -

Bayes' theorem is a fundamental concept in probability theory that plays a crucial role in various machine learning algorithms, especially in the fields of Bayesian statistics and probabilistic modelling. It provides a way to update probabilities based on new evidence or information. In the context of machine learning, Bayes' theorem is often used in Bayesian inference and probabilistic models. The theorem can be mathematically expressed as:

$$P(A/B)=P(B/A)\cdot P(A)/P(B)$$

Where: -

P(A/B) is the posterior probability of event A given event B.

(B/A) is the likelihood of event B given event A.

P(A) is the prior probability of event A.

P(B) is the total probability of event B.

LSTM- Long Short-Term Memory: -

Long Short-Term Memory is an improved version of recurrent neural network designed by Hochreiter & Schmid Huber. A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network. LSTMs model address this problem by introducing a memory cell, which is a container that can hold information for an extended period. LSTM architectures are capable of learning long-term dependencies in sequential data, which makes them well-suited for tasks such as language translation, speech recognition, and time series forecasting. LSTMs can also be used in combination with other neural network architectures, such as Convolutional Neural Networks (CNNs) for image and video analysis.

LSTM Architecture: -

The LSTM architectures involve the memory cell which is controlled by three Gates : the input gate, the forget gate, and the output gate. These gates decide what information to add to, remove from, and output from the memory cell.

- The input gate controls what information is added to the memory cell.
- The forget gate controls what information is removed from the memory cell.
- The output gate controls what information is output from the memory cell.

RBM- Restricted Boltzmann Machine: -

Restricted Boltzmann Machine (RBM) is a type of artificial neural network that is used for unsupervised learning. It is a type of generative model that is capable of learning a probability distribution over a set of input data.

The RBM is trained using a process called contrastive divergence, which is a variant of the stochastic gradient descent algorithm. During training, the network adjusts the weights of the connections between the neurons in order to maximize the likelihood of the training data. Once the RBM is trained, it can be used to generate new samples from the learned probability distribution.

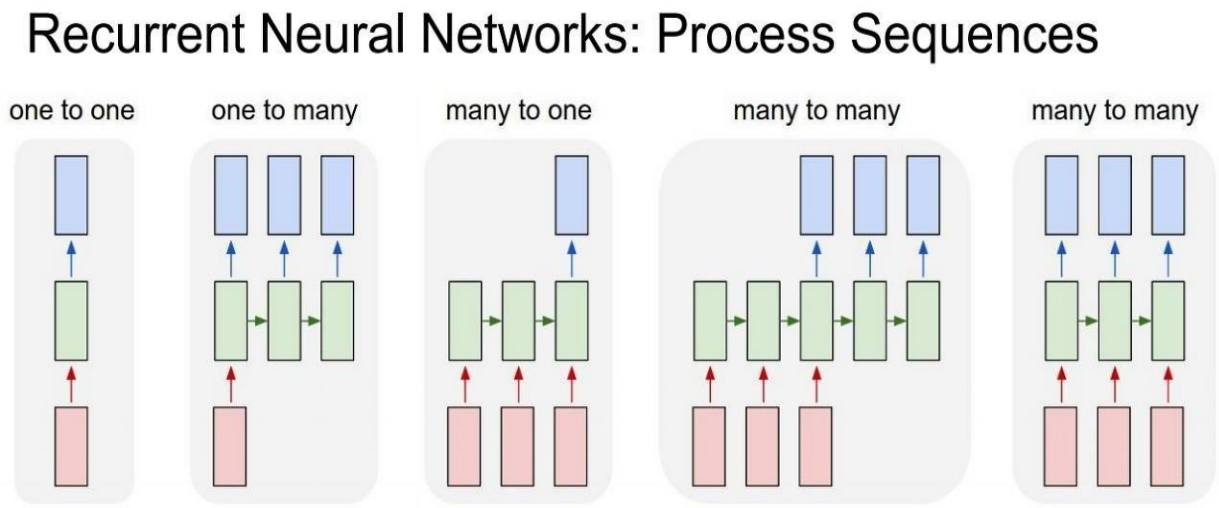
It is a network of neurons in which all the neurons are connected to each other. In this machine, there are two layers named visible layer or input layer and hidden layer. The visible layer is denoted as **v** and the hidden layer is denoted as the **h.** The visible layer actions can be computed by using h = sigmoid (Wv+bh) sigmoid(x) = %(temp(-x)) v = sigmoid (Wx th+b-v)

This can be done using a technique called constructive divergence which as an approximation to the maximum likelihood.

RNN- Recurrent Neural Networks: -

In RNN we have separate and independent input and output layers which we inefficient the dealing with sequential data hence a new neural network called RNN was introduced to store of previous outputs in the internal memory. these results are then fed into the neural network as input this allows it to use in applications like pattern detection speech recognition, NLP, time series prediction. RNN has hidden layers that acts memory locations to store the output of a layer in a loop. There are 4types in recurrent neural network

1. one to one
2. one to many
3. many to one
4. many to many



one to one:

In RNN is one to one which allows a single input & single output. It has fired input and output sizes and acts as a Traditional Neural Networks Applications: Image classification input --> | | --> output

one to many:

One to many is a type of RNN that gives multiple outputs which we given single input. It tables a fixed size and give a sequential of data inputs and the main applications are found in music generation and image capturing.

many to one:

Many-to-one is used when a single output is required from multiple inputs in Sequence. It takes a sequence of inputs to display fixed output.

many to many:

It is used to generate the sequence of output data from a sequence of input data. They are divided into 2 sub categories

1. equal unit size
2. unequal unit size **Equal unit size:**

The no. of both inputs and outputs is same. it can be found in batch normalization and name-entity recognition.

input1 --> | | --> output1

input2 --> | | --> output2

input2 --> | | --> output3

Unequal unit size:

Input and output have different unit numbers and it applications can be found in

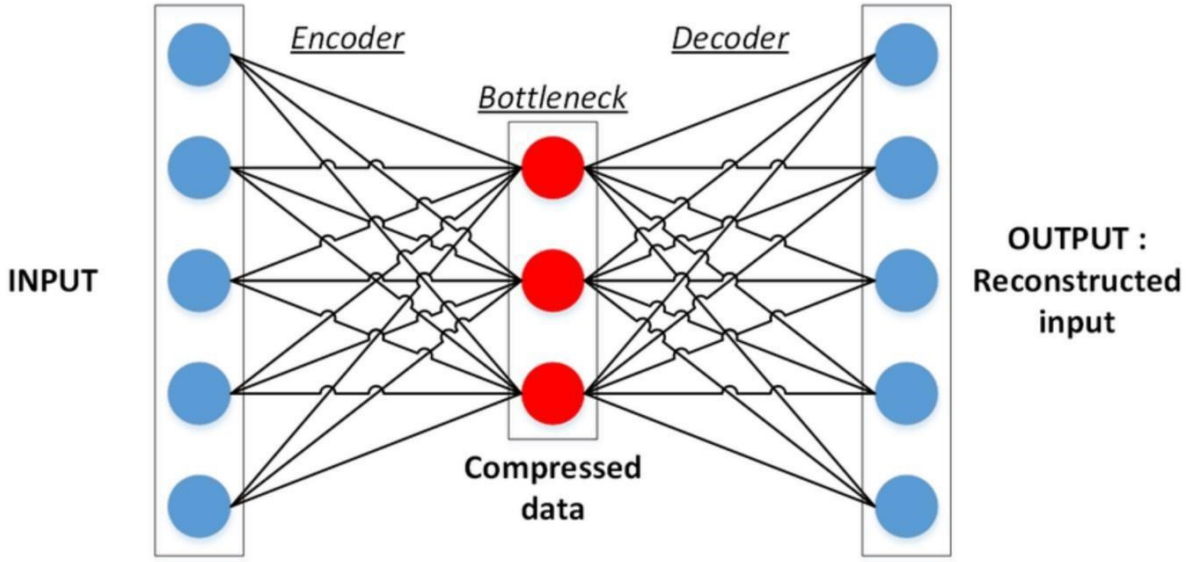
machine translation. input1 --> | | --> output1 input2 --> | | --> output2

Input and output values have different unit numbers

Auto Encoders & Types: -

- An auto encoder is a type of neural network architecture that is used in unsupervised learning.
- The main goal of auto encoder is to learn a component representation of the original data.
- Auto encoders consist of two parts: - Encode, Decode.
- An encoder maps encoding back to original input.
- Decoder maps encoding to back for dimensional encoder.
- The network architecture for auto encoders can vary b/w a simple feed

  Forward network, LSTM, convolutional neural network depending upon user case.



Types of auto encoders:

1. Vanila autoencoders
2. Convolutional auto encoders
3. Recurrent auto encoders
4. Variational auto encoders
5. Denoising auto encoders
6. Adversarial auto encoders.

VGG NET: -

VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The “deep” refers to the number of layers with VGG-16 or VGG-19 consisting of 16 and 19 convolutional layers.

The VGG architecture is the basis of ground-breaking object recognition models. Developed as a deep neural network, the VGG Net also surpasses baselines on many tasks and datasets beyond ImageNet. Moreover, it is now still one of the most popular image recognition architectures.

VGG Architecture: -

VGG architecture consist of blocks, whereas each block of composed of convolution and max pooling layers.

VGG net comes in 2 flavours, vgg16 and vgg19, where as they consist no. of layers in each of them. It is the basic of ground breaking object recognition model.

The architecture has 22 layers with 60 million parameters are used in ReLu (rectified linear unit) activation function. This architecture uses techniques such as |x| convolution in middle of architecture and global average pooling.

--> conv l-1 --> conv l2

--> pooling

--> conv l-1 --> conv l2

--> pooling

--> conv l-1 --> conv l2

--> pooling

--> dense --> dense --> dense --> output

Google Net & Architecture: -

It is used in deep learning model which is developed by researchers of google and it consist of 22-layers and trained on the image net dataset. it can classify objects into 1000 different categories.

Architecture:

conv 2 --> max pooling --> conv 2 --> max pooling --> inception 3a --> inception 3b --> max pooling --> inception 4a --> inception 4b --> inception 4c --> inception 4d --> inception 4d --> inception 4c --> inception 5a --> inception 5b --> inception 5c -> dropout 40% --> soft max

Data Types in Python: -

Python: -

Python is an interpreted, object-orientated high level programming language. It was created by Gudio Van Rossum in 1991. Python supports modules and packages which encourages program modularity and code reusability. Python works on different platforms such as windows, mac OS, Linux etc. Python has syntax that allows develop to write programs with fewer lines compared to other programming languages. There are 14 data types. They are as follows: -

- INT: -It consists the integer values i.e. numbers. 2, 3, ….
- FLOAT: - It consists of floating values i.e. decimals 0.78, 0,66, ….
- COMPLEX: - It consists of imaginary and real numbers. 2+3i, ….
- CHAR: - It consists of characters i.e. alphabets. A, C, e ….
- STR: -It consists of string values i.e. group of characters. ‘Python’
- BYTE: - Consists of 0 and 1.
- BOOL: -Consists of Boolean type i.e. TRUE and FALSE.

- SET: -Consists the data items in a curly-braces. ,1, 5- • TUPPLE: - Consists the data items in parentheses. (78, 98)
- DICT: -Consists the data items in curly-braces. ,1, 6, 9, 8- • FROZEN SET: - Consists of set Operations. union, intersection
- RANGE: - Consists of range values. for i in range
- LIST: -Consists the data items in square brackets. *3, 6, 8+

Arithmetic Operations in Python:

Arithmetic operators in python: -

Arithmetic operators are used with numeric values to perform common mathematical operations.

Following are the arithmetic operators with examples: -

OPERATOR	NAME	EXAMPLE	X=9 and Y=3
+	Addition	x + y	9 + 3 = 12
-	Subtraction	x - y	9 – 3 = 6
*	Multiplication	x * y	9 * 3 = 27
%	Modulus	x % y	9 % 3 = 0
**	Exponent	x ** y	9 **3= 729
//	Floor division	x // y	9 // 3 = 3

Declaration of Comments & Variables: -

Variables in python: -

Variables are used to store data values. In python, address is also created for variables.

SYNTAX: - variable = value

Declaration of variables: -

While declaring variables, we have to keep some rules in mind. They are: -

- It should not start with digit or symbols
- We can use underscore at first then use digits or symbols.
- Some of the examples are as follows: - _python123, n=50 etc.

Comments in python: -

- Comments are used for the description of the code.
- There are two types of comment lines. They are: -

1. Single line comment

2. Multiple line comment o Single line comments are declared with the symbol (#) in-front of them

o Multiple line comments are declared with triple quotations (‘ ’ ’).

Examples are as follows: - # Python is a high-level language.

‘ ’ ’

Python is interpreted language It executes the code line by line

‘ ‘ ‘

Reserved Words in Python: -

There are 35 reserved words in python. They are: -

F0B7 Boolean constraints:

- True
- False
- None
- Logical operators:
- and
- or
- not
- is
- Conditional statements:

- if
- elif
- else
- Looping statements:
- while
- for
- break
- continue
- return
- in
- yield
- Exception handling:
- try
- expect
- finally
- raise
- assert
- Module and import:
- import
- from
- as
- Function and class definition:
- class
- def
- pass
- global
- nonlocal
- lambda
- del
- Context managers:
- with
- async

- await

Control Statements in Python: -

Control statements: -

Control statements in python are used to control the flow of execution of a program.

The three types of control statements are: -

1. Conditional statements
2. Jumping statements
3. Looping statements

- Conditional statements: -

if statement: -

The `if` statement executes a block of code only if a specified condition is true.

SYNTAX: -

if condition: statement

if...else statement: -

The `if…else` statement allows you to execute the instructions if one block of code condition is true, and another block when if the condition is false.

SYNTAX: -

```
if condition:
        statement
else:
        statement
```

if...elif...else statement: -

The `if…elif…else` statement allows you to check multiple conditions and execute different blocks of code based on which condition is true.

SYNTAX: -

if condition1:

statement elif

condition2:

statement

else:

statement

Nested if statement:

Nested `if` statement is an `if` statement that is placed inside another `if` statement.

SYNTAX: -

if condition1:

if condition2:

statement

Nested if...else statement: -

Nested `if…else` statement is similar to a nested `if` statement, but it includes an `else` block for each `if` condition.

SYNTAX: -

if condition1:

if condition2:

statement1
else:
statement2
else:

statement3

- Jumping statements: -

Break statement: -

The break statement terminates the loop it is currently in, regardless of whether the loop condition is true or false.

EXAMPLE: -

for i in range (10):

If i==5:

Break

Print(i)

Continue statement: -

The continue statement skips the rest of the code inside the loop for the current iteration and proceeds to the next iteration of the loop.

EXAMPLE: -

```
for i in range (10):
if i==5: continue
print(i)
```

Pass statement:

-

The pass statement is a null operation; nothing happens when it is executed.

EXAMPLE: - for

```
i in range
(5): if
i==3:
pass
else:
print(i)
```

- Looping statements: -

While loop: -

A while loop is used to execute a block of statements repeatedly until a given condition is satisfied.

SYNTAX: - while

```
expression:
statement
```

For loop: -

It can be used to iterate over a range and iterators. SYNTAX: -

```
for iterator_var in range:

statement
```

Nested loop: -

Python programming language allows to use one loop inside another loop which is called nested loop. SYNTAX1: - for iterator_var in sequence: for iterator_var in sequence:

statements(s)

statements(s)

SYNTAX2: -

while expression:

while expression:

statement(s)

statement(s)

Programs:-

Given number is positive or negative or zero-

```
n=float(input())
if n>0: print(“positive”)
elif n<0: print(“negative”)
else:
        print(“zero”)
```

Given number is odd or even-

```
n=int(input())
if n%2==0:
        print(“even”) else:
        print(“odd”)
```

Given number is Armstrong or not-

```
n=int(input()) temp=n s=0 while n!=0:
r=n%10 n=n//10 s=s+r*r*r
if s==temp:
        print(‘Armstrong’) else
        print(‘Not Armstrong’)
```

Eliminate Duplicate Values-

```
l=list(map(int,input().split())) u=[ ]
d=[ ]
for i in l:
        if i not in u:
            u.append(i)
        else:
            d.append(i)
for i in u:
        print(i,end=" ")
```

PROBLEM STATEMENT AND EXPLANATION

In a fraud detection project, the goal is to identify fraudulent transactions or behaviors within a dataset. This is crucial for sectors like finance, e-commerce, and insurance where detecting fraudulent activities can prevent significant financial losses and protect users. The dataset for such a project typically contains historical records of transactions or interactions that may be either fraudulent or legitimate.

Example Fraud Detection Dataset Features:

Transaction ID: A unique identifier for each transaction, which helps in tracking and referencing specific records.

Date and Time: The timestamp of the transaction, which can be used to identify patterns over time and detect anomalies.

Amount: The value of the transaction, which can be indicative of fraudulent behavior (e.g., unusually large transactions).

Merchant ID: Identifies the business or merchant involved in the transaction. This can help in analyzing transactions across different merchants and spotting anomalies.

Customer ID: Unique identifier for the customer, which is useful for tracking behavior patterns and identifying unusual activity.

Transaction Type: The type of transaction (e.g., purchase, withdrawal, transfer). Different transaction types may have different fraud patterns.

Location: The geographical location where the transaction occurred. Unusual locations can be a red flag.

Device Information: Details about the device used for the transaction, which may include device ID, operating system, and browser type. 36

Transaction Method: Indicates whether the transaction was made online, in person, via mobile app, etc. Different methods can have different fraud risks.

Customer Behavior: Historical data on customer behavior, such as past transactions and patterns, which helps in establishing normal behavior profiles.

Libraries Used for Analysis and Modeling:

To handle and analyze fraud detection data, as well as to build and evaluate predictive models, you would typically use Python and a range of specialized libraries:

Pandas: For data manipulation and preprocessing tasks, including loading, cleaning, and transforming datasets. Essential for handling large volumes of transactional data.

NumPy: For numerical operations and array manipulations, often used alongside Pandas to perform mathematical computations.

Matplotlib and Seaborn: For data visualization, allowing you to create plots and charts to explore transaction distributions, trends, and identify outliers.

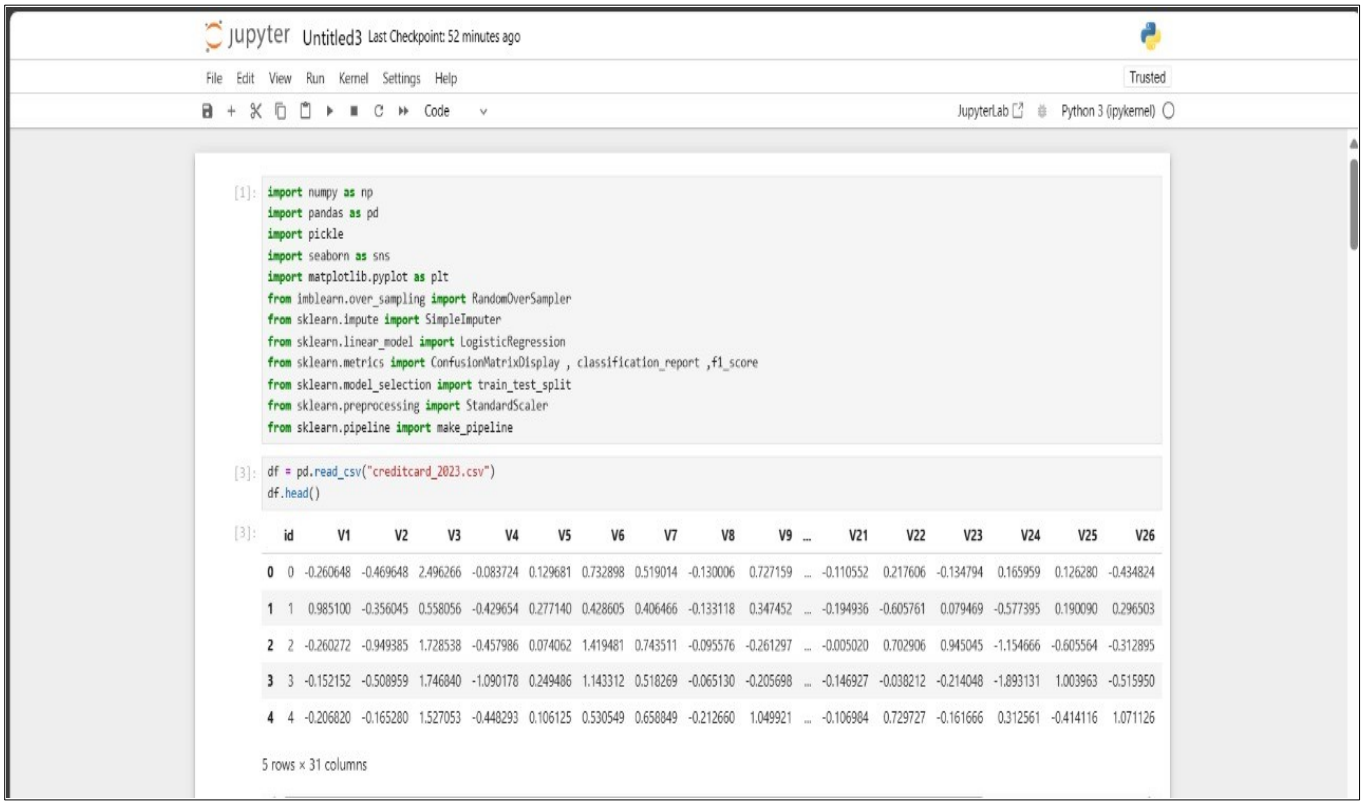
scikit-learn: A versatile library for machine learning tasks, offering tools for preprocessing data, building models (such as classification and anomaly detection), and evaluating model performance.

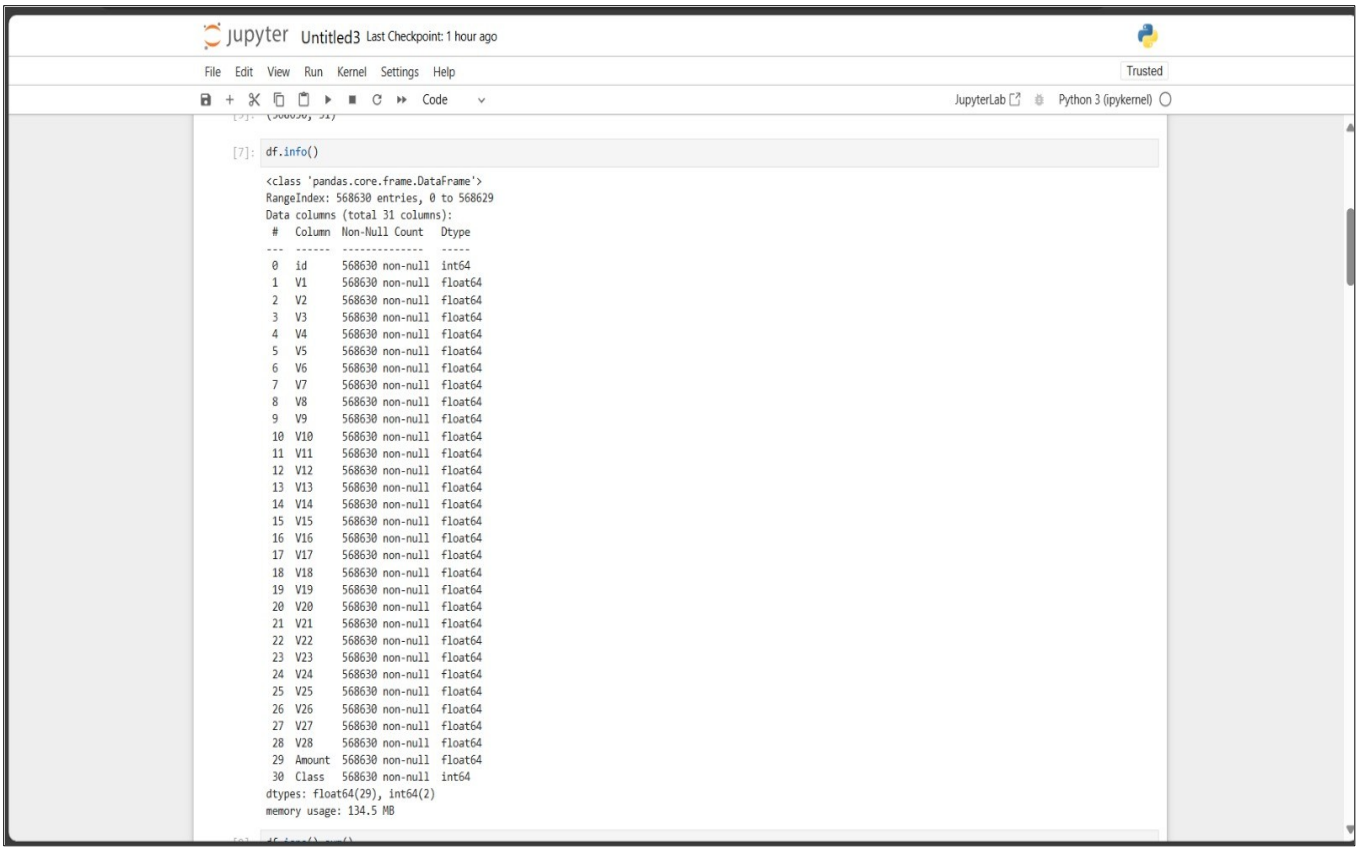
TensorFlow or PyTorch: For advanced machine learning and deep learning models, which can be particularly useful for complex fraud detection scenarios involving large datasets or sophisticated patterns.

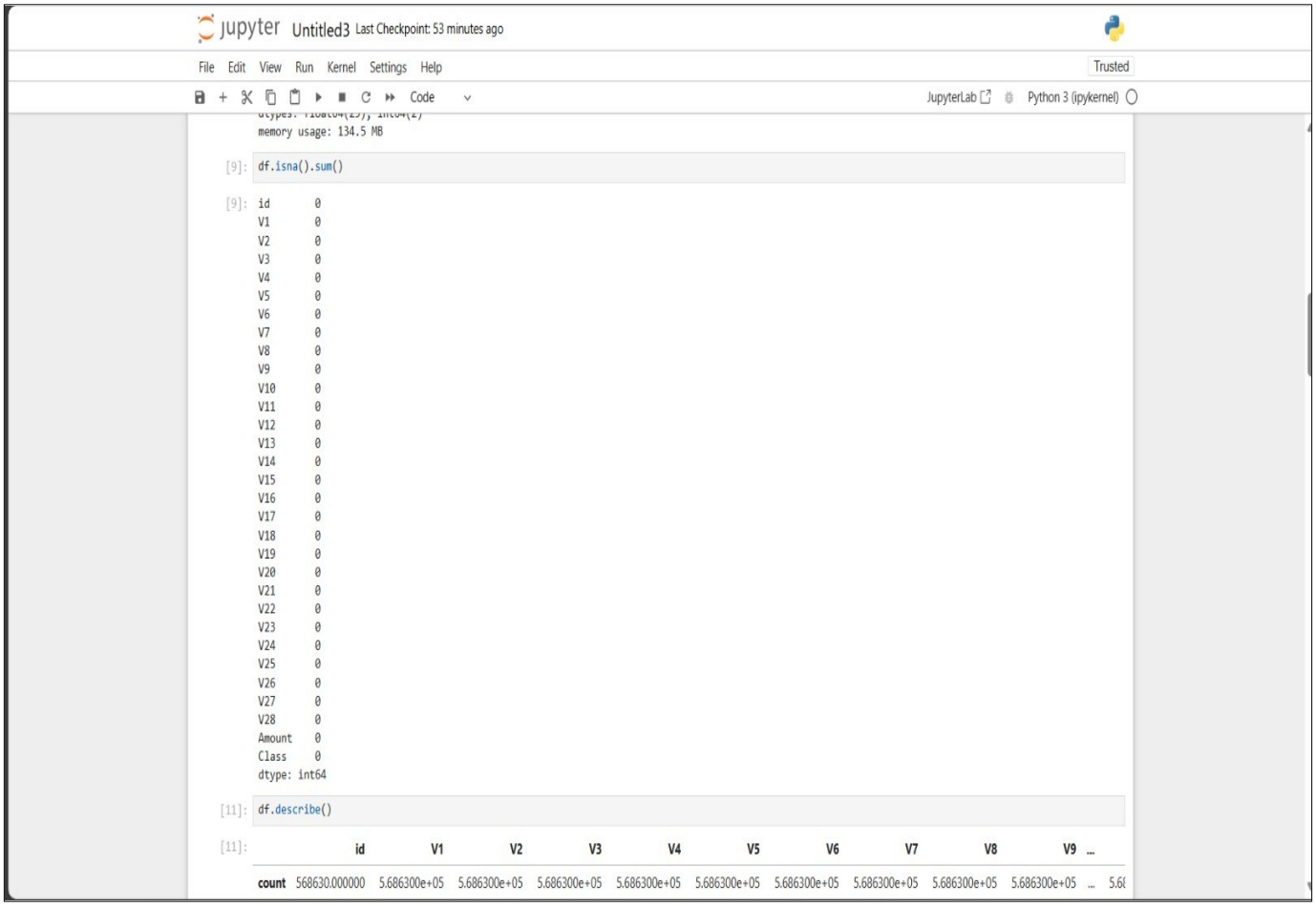
Isolation Forest or One-Class SVM: Specialized algorithms for anomaly detection, which are often used in fraud detection to identify outliers that deviate from the norm.

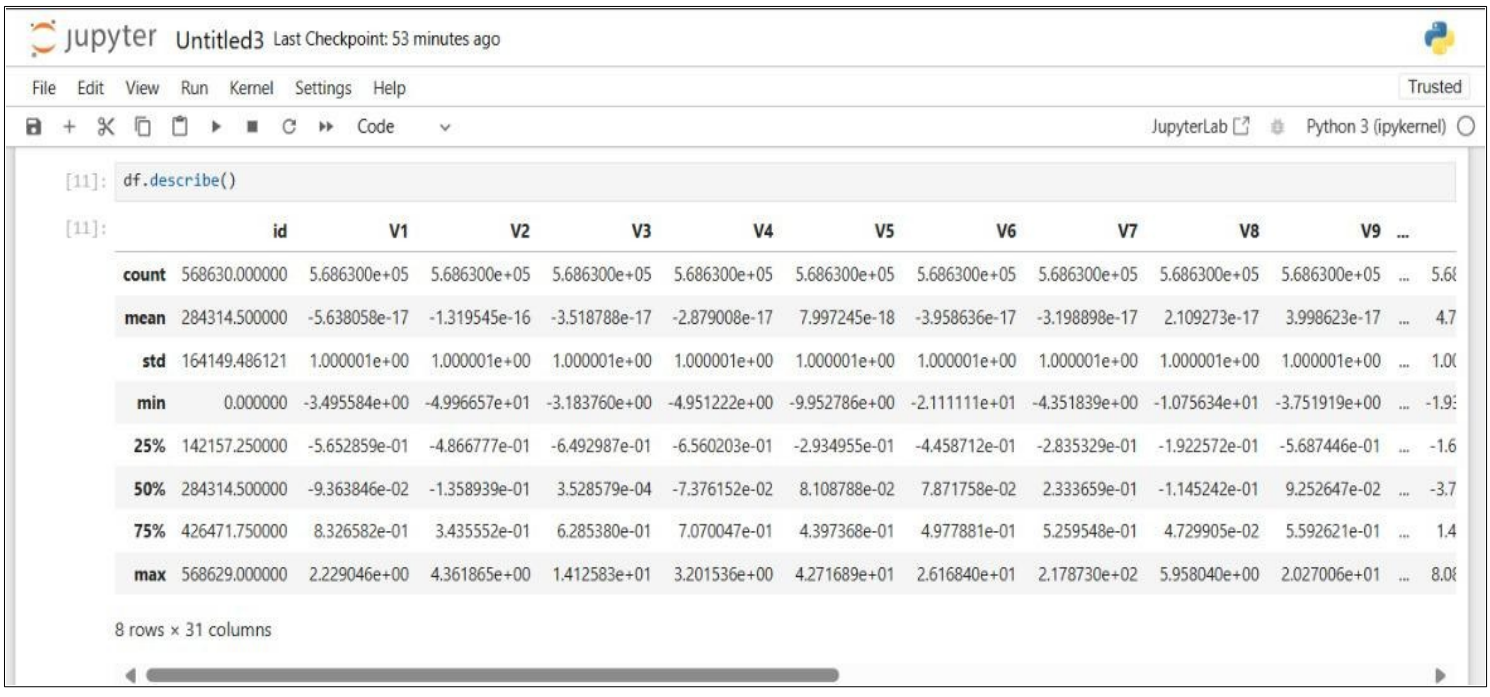
XGBoost or LightGBM: Gradient boosting libraries that can improve prediction performance by combining the output of multiple models, which is beneficial in handling imbalanced datasets typical in fraud detection scenarios

SOURCE AND OUTPUTS


jupyter Untitled3 Last Checkpoint: 52 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[1]:
import numpy as np
import pandas as pd
import pickle
import seaborn as sns
import matplotlib.pyplot as plt
from imblearn.over_sampling import RandomOverSampler
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import ConfusionMatrixDisplay , classification_report ,f1_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
[3]:
df = pd.read_csv("creditcard_2023.csv")
df.head()
[3]:
id V1 V2 V3 V4 V5 V6 V7 V8 V9 ... V21 V22 V23 V24 V25 V26
0 0 -0.260648 -0.469648 2.496266 -0.083724 0.129681 0.732898 0.519014 -0.130006 0.727159 ... -0.110552 0.217606 -0.134794 0.165959 0.126280 -0.434824
1 1 0.985100 -0.356045 0.558056 -0.429654 0.277140 0.428605 0.406466 -0.133118 0.347452 ... -0.194936 -0.605761 0.079469 -0.577395 0.190090 0.296503
2 2 -0.260272 -0.949385 1.728538 -0.457986 0.074062 1.419481 0.743511 -0.095576 -0.261297 ... -0.005020 0.702906 0.945045 -1.154666 -0.605564 -0.312895
3 3 -0.152152 -0.508959 1.746840 -1.090178 0.249486 1.143312 0.518269 -0.065130 -0.205698 ... -0.146927 -0.038212 -0.214048 -1.893131 1.003963 -0.515950
4 4 -0.206820 -0.165280 1.527053 -0.448293 0.106125 0.530549 0.658849 -0.212660 1.049921 ... -0.106984 0.729727 -0.161666 0.312561 -0.414116 1.071126
5 rows × 31 columns


jupyter Untitled3 Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[7]:
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568630 entries, 0 to 568629
Data columns (total 31 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 id 568630 non-null int64
1 V1 568630 non-null float64
2 V2 568630 non-null float64
3 V3 568630 non-null float64
4 V4 568630 non-null float64
5 V5 568630 non-null float64
6 V6 568630 non-null float64
7 V7 568630 non-null float64
8 V8 568630 non-null float64
9 V9 568630 non-null float64
10 V10 568630 non-null float64
11 V11 568630 non-null float64
12 V12 568630 non-null float64
13 V13 568630 non-null float64
14 V14 568630 non-null float64
15 V15 568630 non-null float64
16 V16 568630 non-null float64
17 V17 568630 non-null float64
18 V18 568630 non-null float64
19 V19 568630 non-null float64
20 V20 568630 non-null float64
21 V21 568630 non-null float64
22 V22 568630 non-null float64
23 V23 568630 non-null float64
24 V24 568630 non-null float64
25 V25 568630 non-null float64
26 V26 568630 non-null float64
27 V27 568630 non-null float64
28 V28 568630 non-null float64
29 Amount 568630 non-null float64
30 Class 568630 non-null int64
dtypes: float64(29), int64(2)
memory usage: 134.5 MB


jupyter Untitled3 Last Checkpoint: 53 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
memory usage: 134.5 MB
[9]: df.isna().sum()
[9]: id 0
V1 0
V2 0
V3 0
V4 0
V5 0
V6 0
V7 0
V8 0
V9 0
V10 0
V11 0
V12 0
V13 0
V14 0
V15 0
V16 0
V17 0
V18 0
V19 0
V20 0
V21 0
V22 0
V23 0
V24 0
V25 0
V26 0
V27 0
V28 0
Amount 0
Class 0
dtype: int64
[11]: df.describe()
[11]: id V1 V2 V3 V4 V5 V6 V7 V8 V9 ...
count 568630.000000 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 ... 5.6


jupyter Untitled3 Last Checkpoint: 53 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab Python 3 (ipykernel)
[11]: df.describe()
[11]:
id V1 V2 V3 V4 V5 V6 V7 V8 V9 ...
count 568630.000000 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 5.686300e+05 ... 5.6
mean 284314.500000 -5.638058e-17 -1.319545e-16 -3.518788e-17 -2.879008e-17 7.997245e-18 -3.958636e-17 -3.198898e-17 2.109273e-17 3.998623e-17 ... 4.7
std 164149.486121 1.000001e+00 1.000001e+00 1.000001e+00 1.000001e+00 1.000001e+00 1.000001e+00 1.000001e+00 1.000001e+00 1.000001e+00 ... 1.0
min 0.000000 -3.495584e+00 -4.996657e+01 -3.183760e+00 -4.951222e+00 -9.952786e+00 -2.111111e+01 -4.351839e+00 -1.075634e+01 -3.751919e+00 ... -1.9
25% 142157.250000 -5.652859e-01 -4.866777e-01 -6.492987e-01 -6.560203e-01 -2.934955e-01 -4.458712e-01 -2.835329e-01 -1.922572e-01 -5.687446e-01 ... -1.6
50% 284314.500000 -9.363846e-02 -1.358939e-01 3.528579e-04 -7.376152e-02 8.108788e-02 7.871758e-02 2.333659e-01 -1.145242e-01 9.252647e-02 ... -3.7
75% 426471.750000 8.326582e-01 3.435552e-01 6.285380e-01 7.070047e-01 4.397368e-01 4.977881e-01 5.259548e-01 4.729905e-02 5.592621e-01 ... 1.4
max 568629.000000 2.229046e+00 4.361865e+00 1.412583e+01 3.201536e+00 4.271689e+01 2.616840e+01 2.178730e+02 5.958040e+00 2.027006e+01 ... 8.0
8 rows × 31 columns

jupyter Untitled3 Last Checkpoint: 53 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

75%	426471.750000	8.326582e-01	3.435552e-01	6.285380e-01	7.070047e-01	4.397368e-01	4.977881e-01	5.259548e-01	4.729905e-02	5.592621e-01	... 1.4
max	568629.000000	2.229046e+00	4.361865e+00	1.412583e+01	3.201536e+00	4.271689e+01	2.616840e+01	2.178730e+02	5.958040e+00	2.027006e+01	... 8.0

8 rows × 31 columns

[53]:
```
import seaborn as sns
import matplotlib.pyplot as plt # Import the matplotlib library and give it the alias 'plt'

#corr = df.drop(columns=['Class']).corr()
#sns.heatmap(corr);

plt.rcParams['figure.figsize']= (22,11)

plt.title("Correlation Heatmap",fontsize=18, weight= 'bold')

sns.heatmap(df.corr(), cmap="BuPu", annot=True)

plt.show()
```

jupyter Untitled3 Last Checkpoint: 53 minutes ago

File Edit View Run Kernel Settings Help

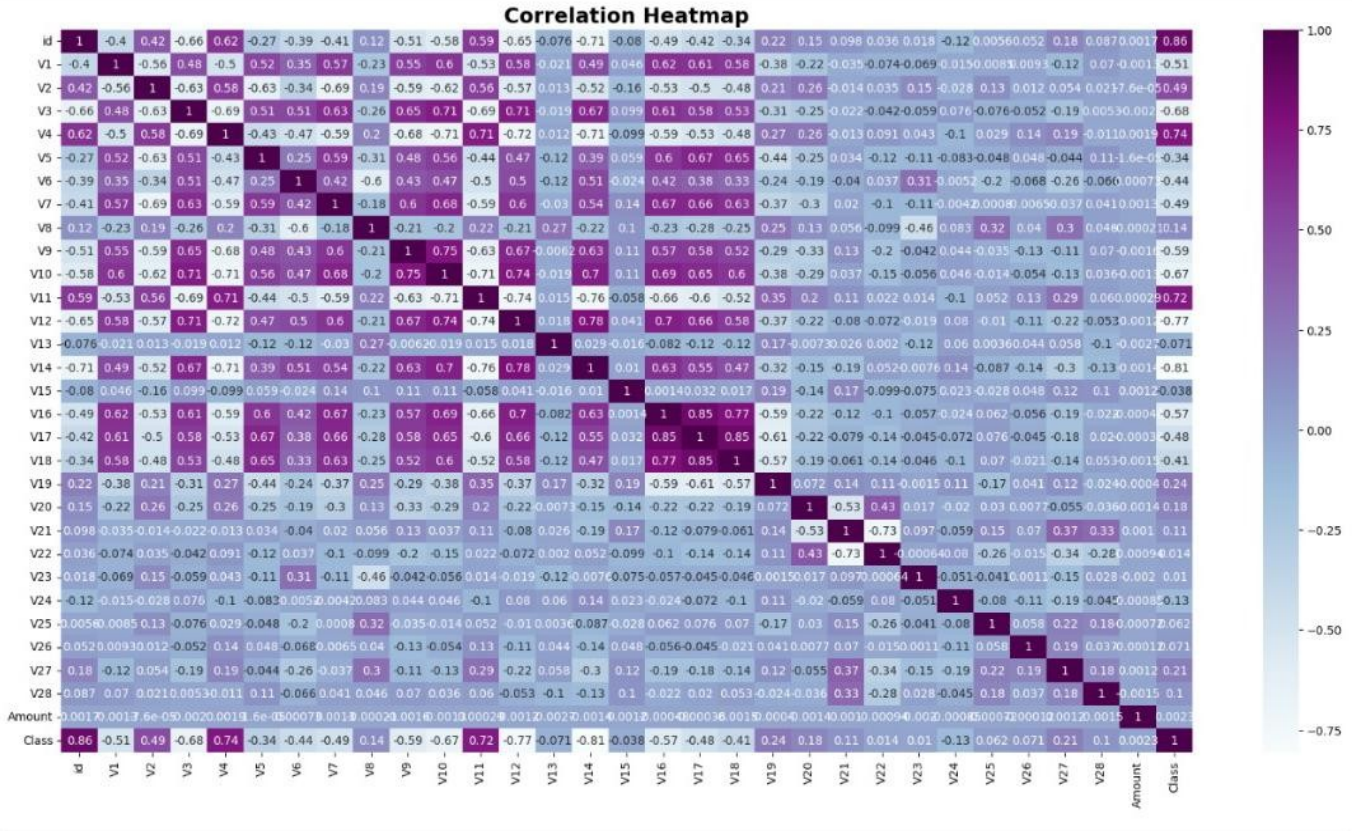
Trusted

Code

JupyterLab Python 3 (ipykernel)

```
sns.heatmap(df.corr(), cmap="BuPu", annot=True)

plt.show()
```



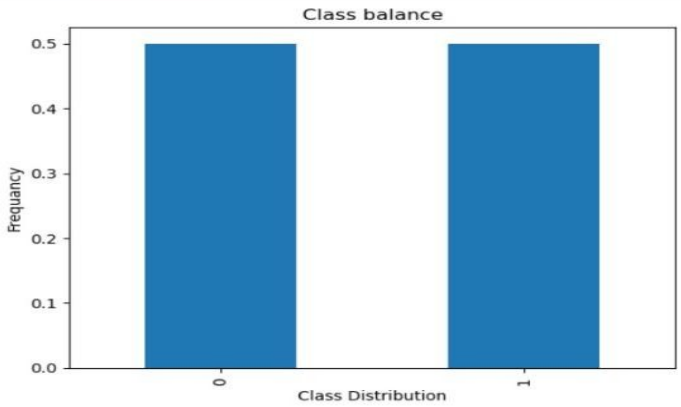
jupyter Untitled3 Last Checkpoint: 53 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

id V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 Amount Class

[13]:
```
df['Class'].value_counts(normalize= True).plot(kind= 'bar')
plt.xlabel("Class Distribution")
plt.ylabel("Frequancy")
plt.title("Class balance");
```



[17]:
```
x= df.drop(['id', 'Class'], axis= 1)
y= df['Class']
```

[19]:
```
stn_scaler = StandardScaler()

x_scaled = stn_scaler.fit_transform(x)
```

jupyter Untitled3 Last Checkpoint: 53 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

Class Distribution

[17]:
```
x= df.drop(['id', 'Class'], axis= 1)
y= df['Class']
```

[19]:
```
stn_scaler = StandardScaler()

x_scaled = stn_scaler.fit_transform(x)
```

[21]:
```
X = pd.DataFrame(x_scaled,columns=x.columns)
```

[23]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print("X shape:", X.shape)
print("y shape:", y.shape)
```

```
X shape: (568630, 29)
y shape: (568630,)
```

[25]:
```
print("X_train shape:", X_train.shape)
print("y_train shape:", y_train.shape)
print("X_test shape:", X_test.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (454904, 29)
y_train shape: (454904,)
X_test shape: (113726, 29)
y_test shape: (113726,)
```

[27]:
```
acc_baseline = y_train.value_counts(normalize=True).max()
print("Baseline Accuracy:", round(acc_baseline, 4))
```

```
Baseline Accuracy: 0.5002
```

[29]:
```
clf = LogisticRegression()
```

[31]:
```
clf.fit(X_train , y_train)
```

[31]:
```
LogisticRegression()
```

jupyter Untitled3 Last Checkpoint: 53 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[33]:
```
clf.predict(X_train)
```

[33]:
```
array([1, 1, 1, ..., 1, 0, 0], dtype=int64)
```

[35]:
```
acc_train = clf.score(X_train , y_train)
acc_test = clf.score(X_test , y_test)

print(f"Training accuracy: {round(acc_train , 4)}")
print(f"test accuracy: {round(acc_test , 4)}")
```

```
Training accuracy: 0.9649
test accuracy: 0.9652
```

jupyter Untitled3 Last Checkpoint: 54 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[39]:
```
print(classification_report(

    y_test,
    clf.predict(X_test)

))
```

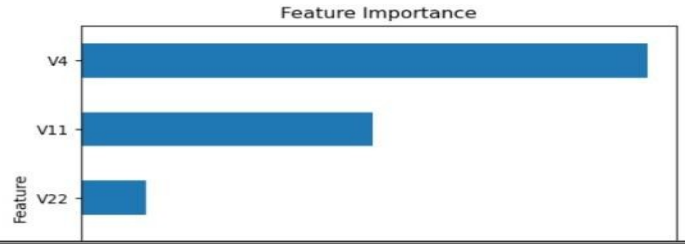
```
              precision    recall  f1-score   support

           0       0.95      0.98      0.97     56750
           1       0.98      0.95      0.96     56976

    accuracy                           0.97    113726
   macro avg       0.97      0.97      0.97    113726
weighted avg       0.97      0.97      0.97    113726
```

[41]:
```
features = X_test.columns
importances = clf.coef_[0]
```

[43]:
```
feat_imp = pd.Series(importances , index=features).sort_values()
feat_imp.tail().plot(kind= 'barh')
plt.xlabel("Scale Importance")
plt.ylabel("Feature")
plt.title("Feature Importance");
```



jupyter Untitled3 Last Checkpoint: 54 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[43]:
```
feat_imp = pd.Series(importances , index=features).sort_values()
feat_imp.tail().plot(kind= 'barh')
plt.xlabel("Scale Importance")
plt.ylabel("Feature")
plt.title("Feature Importance");
```