A SHORT-TERM INTERNSHIP REPORT ON

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

BY

SIRIGISETTI NIRANJANSAI

III DATA SCIENCE

Under the Esteemed Guidance of

Mr. G.V.S.S PRASANTH SIR

Tutor of Artificial Intelligence & Machine Learning

ADITYA DEGREE COLLEGE, TUNI

(Affiliated to ADIKAVI NANNAYA UNIVERSITY)

Tuni-533401, Kakinada District, ANDHRA PRADESH

2022-2025

ADITYA DEGREE COLLEGE

DECLARATION BY THE STUDENT

I hereby declare that the work described in this Short-term Internship, entitled "**Artificial Intelligence & Machine Learning**" which is being submitted by me in partial fulfilment of the requirements for the award of degree of **Bachelor of Science** from the Department of **Data Science** to Aditya Degree College,Tuni under the guidance of **Mr. G.V.S.S PRASANTH Sir** tutor of **Artificial Intelligence & Machine Learning** in Aditya Degree College , Tuni.

Place:TUNI **(NIRANJANSAI SIRIGISETTI)**

Date:

ADITYA DEGREE COLLEGE

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the Short Term Internship entitled, "**ARTIFICIAL INTELLIGENCE & MACHINE LEARNING",** that is being submitted by **NIRANJANSAI SIRIGISETTI** bearing **221177156187** of **III Data Science** , which is being submitted by us in partial fulfilment of the requirements for the award of degree of **Bachelor of Science** from the Department of Data science to Aditya Degree College, bonified work carried out by him under my guidance and Supervision.

(Mr. G.V.S.S PRASANTH SIR)

ACKNOWLEDGEMENT

No endeavor is completed without the valuable support of others.
I would like to take this opportunity to extend my sincere gratitude to all those who have contributed to the successful completion of this ShortTerm Internship Project Report.

I express my deep sense of gratitude to **SMT. M. DEEPTHI, Principal**, for his Efforts and for giving us permission for carrying out this Long-Term Internship.

I feel deeply honored in expressing my sincere thanks to Mr. G.V.S.S Prashanth Sir tutor of ULearn Visakhapatnam for making the resources available at right time and providing valuable insights leading to the successful completion of my short-Term Internship Project Report.

Finally, I thank all the faculty members of our department who contributed their valuable suggestions in completion of Short-Term Internship report and I also put my sincere thanks to My Parents who stood with me during the whole Short-Term Internship.

(S.NIRANJANSAI)

INTRODUCTION

Credit card fraud detection is a set of methods and techniques designed to block fraudulent purchases, both online and in-store. This is done by ensuring that you are dealing with the right cardholder and that the purchase is legitimate.

When it comes to identifying the cardholder, credit card fraud detection relies on authentication techniques such as MFA (multi-factor authentication), 3DS, biometrics, and OTP (one-time passwords).

However, it is also possible to detect credit card fraud by looking at anomalies in the transaction. For instance, an IP address could point to a suspicious geolocation. Similarly, a device with a never-seen configuration of software and hardware could raise red flags.

Depending on the kind of detection tools your company uses, you may answer questions about the cardholder identity and intention in real-time or retroactively. In that sense, credit card fraud detection can be either a payment fraud prevention measure or a way to investigate previous transactions.

Credit card fraud detection using **Machine learning** (ML) is a vital application in the financial sector, aimed at identifying and preventing fraudulent transactions. Credit card fraud can result in significant financial losses for both consumers and financial institutions. The growing volume of online transactions has made fraud detection more critical than ever.

Machine learning (ML) and artificial intelligence (AI) are pivotal in credit card fraud detection, enabling real-time analysis and identification of suspicious transactions. By analyzing vast amounts of transaction data, ML models can recognize patterns and anomalies that indicate potential fraud. Supervised learning techniques, such as logistic regression and neural networks, use labeled data to predict fraudulent activities, while unsupervised methods, like clustering and autoencoders, detect outliers in the transaction data. These models continuously learn and adapt to new fraud tactics, ensuring robust protection against evolving threats. Additionally, AI-driven decision-making systems provide real-time alerts, enhancing the efficiency and accuracy of fraud prevention measures, thereby safeguarding consumers and financial institutions from significant financial losses.

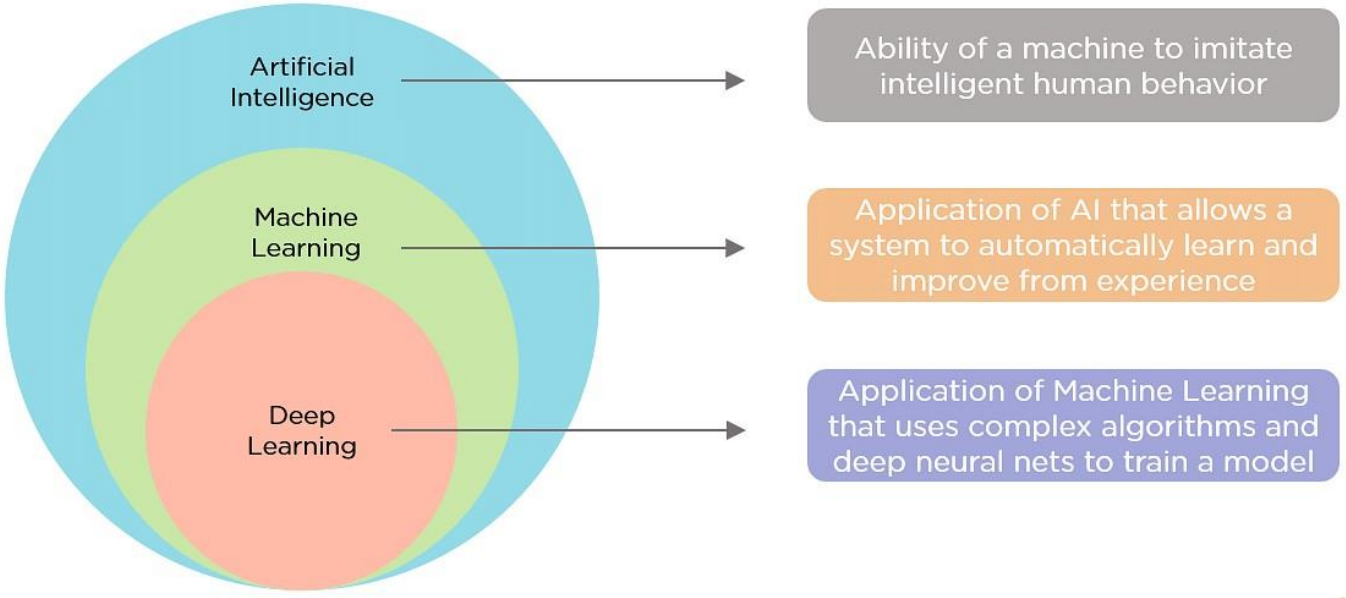
LEARNING OUTCOME OF SHORT TERM INTERNSHIP

What is AI?

- Ai is an branch of computer science it is the simulation of human intelligence process by machines is called AI
- Human intelligence which performs machines is called AI

In AI how many subsets are there?

- Machine learning
- Deep learning



Sub fields

- Artificial neural networks
- Cognative computing
- Natural language processing

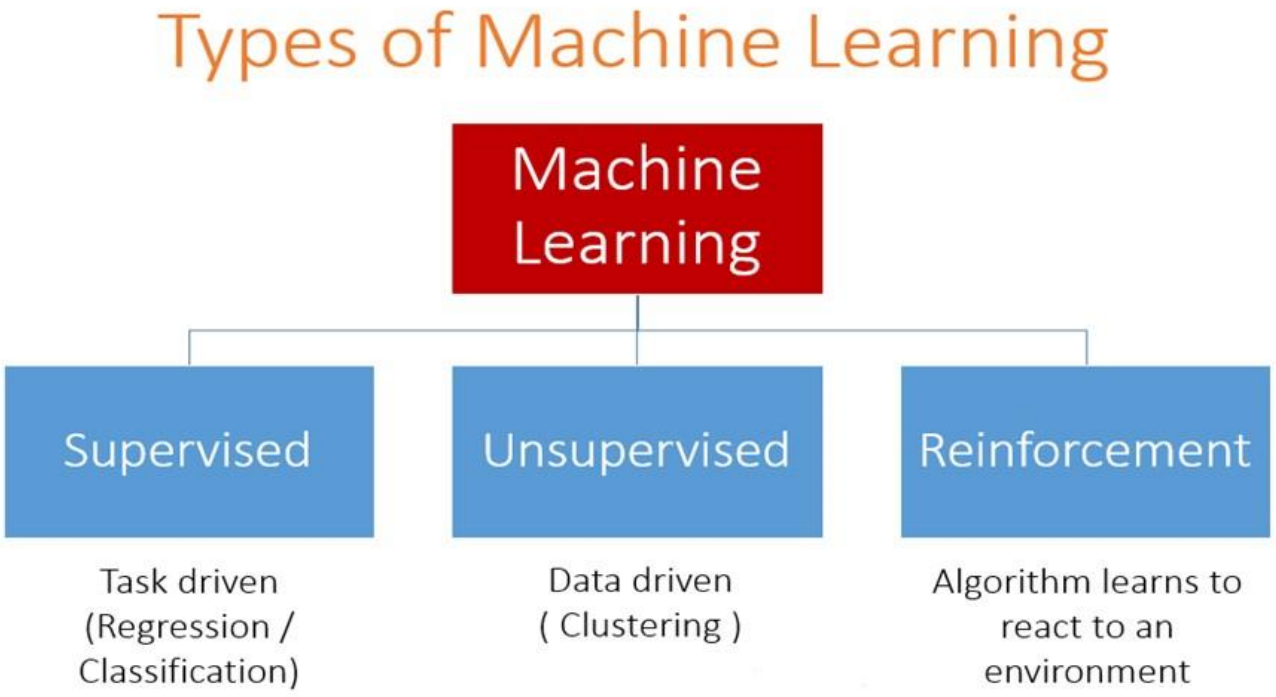
What is Machine learning?

- Its a subset of ai which focus on the use of data and algorithms imitate the way that human learn and fradually increasing its accuracy
- It learns from data & solve the problems

Types of machine learning:

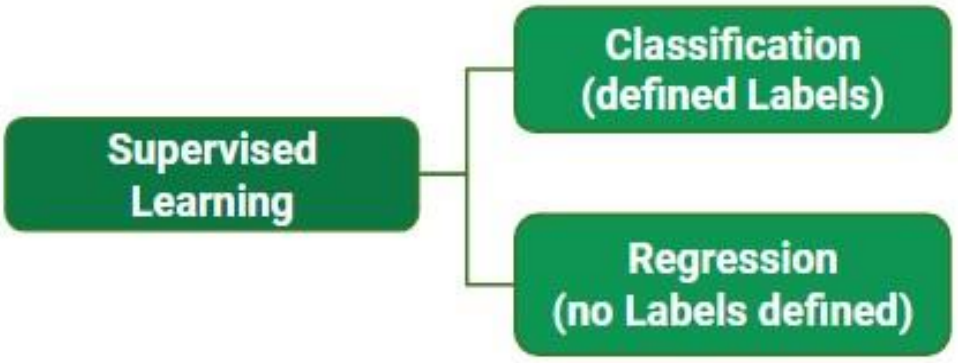
There are 3types of ML

- Supervised learning - its an labelled data or structered data
- Unsupervised learning - its an unlabelled data or unstructered data
- Reinforcement learning -it uses both structured data and unstructured data



SUPERVISERED LEARNING:

- Classification
- Regression



Classification:

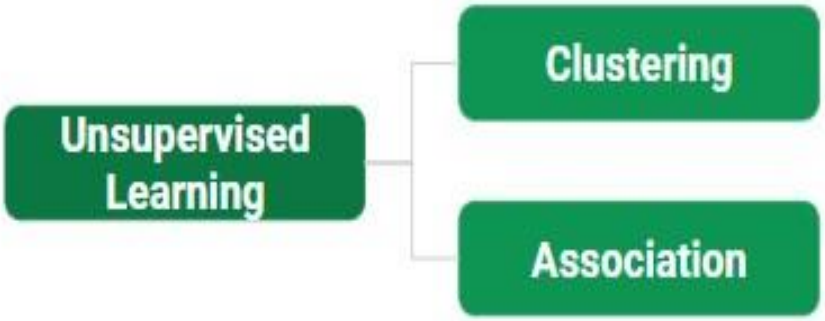
- KNN
- Support vector Machine
- Decision Tree
- Random Forest
- Navie Bays
- Neural Network
- Stochastic Gradient Descent

Regression:

- Linear regression
- Logistic Regression
- Polynomial Regression

UNSUPERVISED LEARNING:

- Clustering
- Association



Clustering:

- Principal Component Analysis
- Hierarchical Clustering
- Singular Value Decomposition
- Independent Component Analysis
- K-Means Clustering

Reinforcement Learning:

3 Ways of implementation

- Model based
- Policy based
- Value based

Models:

- Q-learning
- Markov Models

Programming Languages used in ML:

- Python
- R Language

Python:

Python is one of the easiest yet most useful programming languages which is widely used in the software industry. People use Python for Competitive Programming, Web Development, and creating software. Due to its easiest syntax, it is recommended for beginners who are new to the software engineering field. Its demand is growing at a very rapid pace due to its vast use cases in Modern Technological fields like Data Science, Machine learning, and Automation Tasks.

Mathematics:

- Linear Algebra
- Calculas
- Probability

Data Bases:

To store the structured or unstructured data in database to access the data by using Sql queries (structured query language)

- MangoDB
- My SQL

Visualization Tools:

- Qlick Sense
- Tableau
- PowerBI

Why we used this bi tools?

For data visulatation purpose

Bar Graphs

Pie charts

Histograms

Box plots

Scatterplot

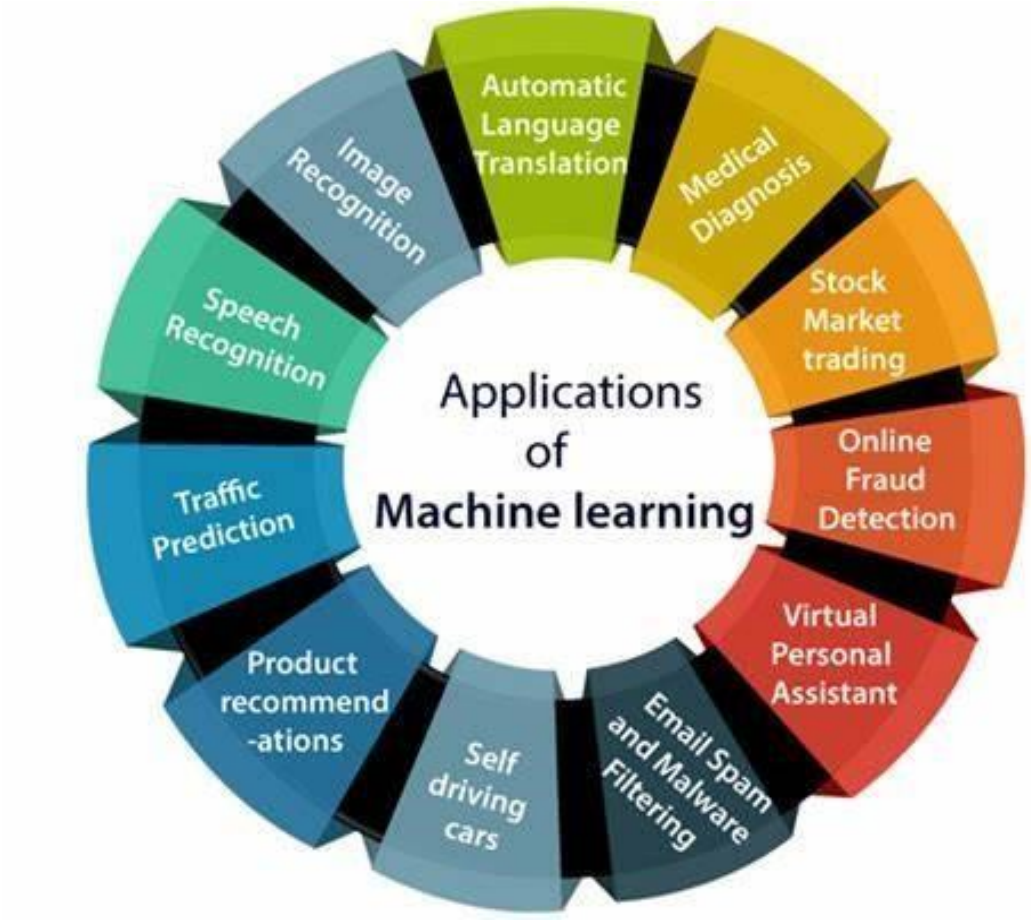
Line charts

Steps required to train AI And ML models:

- Gathering Data
- Data prepration
- Choosing A model
- Training
- Evaluation
- Parameter tuning
- Prediction

Applications of Machine Learning:

- Self-Driving Cars
- Chatbot
- Image registration
- Speech Recognition
- Stock Market trading
- Email Spam Filtering
- Online fraud Detection



AI tools we used in our daily life:

1. Chatbots

A prominent example of this is **ChatGPT**. People started out using this chatbot just another online companion. However, you'll be surprised to know that, ChatGPT is actually an **artificial intelligence-powered chatbot.**

2. Microsoft Bing

While Google has always been the go-to search engine for almost everyone, Microsoft has now revamped Bing with artificial intelligence. The new AI Bing has been specially created to give the search engine the power to intelligently give nuanced responses by its AI. However, Bing also benefits from the new Chat mode.

3.Smart Compose, Quick Reply, and Grammar Check

If you use Gmail then you might have noticed a feature called Smart Compose. It suggests complete sentences based on the preceding line that you have written. It uses Artificial Intelligence to quickly compose your **email drafts with contextual accuracy** and correct grammar. I use it quite often and believe me, it's pretty helpful. There could be no better example of AI making life better and saving time on the other hand.

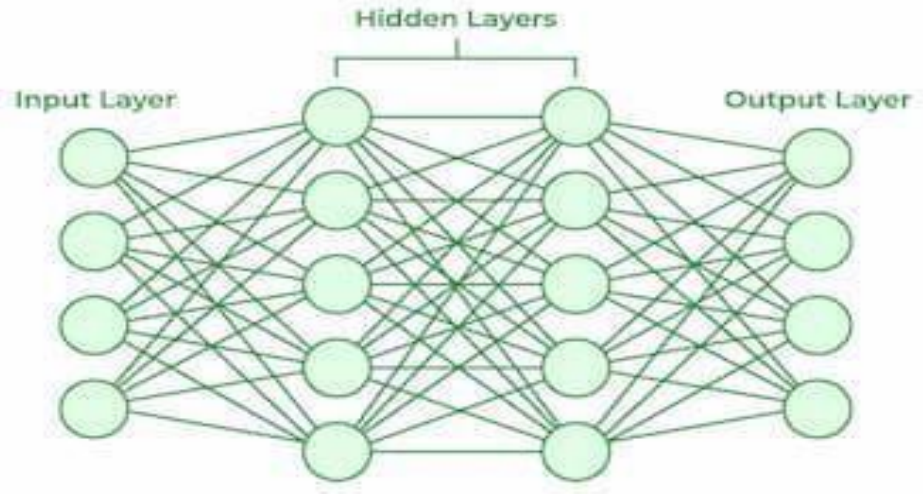
4.Google Lens and OCR

Google Lens is another Google service that is built on AI and has some great technology for fast and accurate optical recognition. It allows you to search for anything through images. Just **point the camera to a shoe or a plant** or an animal or a

text, and it can detect the type of subject and will provide precise information on that thing in just a few seconds.

What is neural Network?

Neural Networks are computational models that mimic the complex functions of the human brain. The neural networks consist of interconnected nodes or neurons that process and learn from data, enabling tasks such as pattern recognition and decision making in machine learning. The article explores more about neural networks, their working, architecture and more.



Input Layer: Each feature in the input layer is represented by a node on the network, which receives input data.
Weights and Connections: The weight of each neuronal connection indicates how strong the connection is. Throughout training, these weights are changed.
Hidden Layers: Each hidden layer neuron processes inputs by multiplying them by weights, adding them up, and then passing them through an activation function. By doing this, non-linearity is introduced, enabling the network to recognize intricate patterns.
Output: The final result is produced by repeating the process until the output layer is reached.

Back propagation:

- In machine learning, backpropagation is an effective algorithm used to train artificial neural networks, especially in feed-forward neural networks.
- Backpropagation is an iterative algorithm, that helps to minimize the cost function by determining which weights and biases should be adjusted. During every epoch, the model learns by adapting the weights and biases to minimize the loss by moving down toward the gradient of the error. Thus, it involves the two most popular optimization algorithms, such as gradient descent or stochastic gradient descent.

Objection Detection:

Object detection is a technique that uses neural networks to localize and classify objects in images. This computer vision task has a wide range of applications, from medical imaging to self-driving cars.

Applications:

- Object reg
- Facial Recg
- Video Tracking
- Moment Detection
- Self Driving Cars
- Animal Detection
- Robotics

Steps involved in objection detection:

- Image preprocessing - resizing and normalization of image
- Feature extraction - it classifies the image
- Object localization - it locates the object
- Object classification - it identifies what type of object in image
-Post processing - it refining and eliminating the duplicate detections.

Gan - generative adversarial networks

-Gan was introduced by LAN goodfellow in 2014.
-Gan are algorithmic architecture that uses two neural networks pitting one against to other to generate new data that passes through the real data.

Applications:

- To generate photorealistic images
- Change facial expressions
- Create computer game scenes
- Visualize designs
- Create artwork

Diff b/w original image and generated images:
-Brightness
-Thickness
-Color
-Background
-Saturation
-unrealistic elements
-Quality
-Clarity
-Size
-Features

DEEP DREAM:

- Deep dream is one of the application of deep learning in computer vision.
- In this deep dream concept we used deep neural networks
- We are using CNN algorithm to find image patterns in images
- Deep dream software original image using deep CNN named as inception

PROCESS:

- If we take any image the deepdream will identify faces and there patterns in image by using deep CNN algorithm to modify the images.
- once we trained this algorithm its reverse process takes place to change the image patterns.
- This can be visualizations to understand the emergent structure of neural network and basis for the deep dream concept

DEEP FAKE:

It is digitally alterated image, video, or audio that replaces one person face with another person face is called deep fake

Algorithms:

- Deep CNN
- GAN Model
- VGG16 and VGG19 - visual geometrical graphs

Training:

Jupyter notebook:

- required libraries
- importing datasets
- selecting input image
- targeted image
- training algorithm
- output

Examples:
Trump face -Putin Face Moffed Girl
Face – Celebrity Face.

DATA AUGMANTATION:

Data augmentation uses pre-existing data to create new data samples that can improve model optimization and generalizability.

- Data Augmantation is data analysis are techniques used to increased the amount of data by adding or slightly modifying copies in already existing data or newly created synthetic data is called data augmantation.
- It is a set of techniques to artificially increase the amount of data generating new data points from existing data

WHY ITS IMPORTANT?

- It includes making small changes to data or using deep learning models to generate new data points.
- it is useful to improve the performance and outcomes of machine learning models by forming new and different examples to train datasets.

STEPS:

- Input data that feed to the data Augmantation pipeline
- The data Augmantation pipeline by sequential steps with different Augmantations
- Tf1-rotation
- Tf2-greyscale to rgb tf3-blur
- Tfn-flip
- The image is feed through the pipeline and processed through each step with different probablity
- After image is processed the human expert randomaly verifies the augmanted results and passes the feedback to the system
- After human expert verification the augmanted data is ready to train the ai training process

FOR IMAGE CLASSIFICATION AND SEGMENTATION:

Random
Rotating re
Scalling
Vertical or horizontal flipping
ranslation Cropping
zooming

PARAMETER SHARING AND TYPING:

It is an convolutional neural network model which is used to share the weights equally in neural networks is called parameter sharing and typing

It is an deep learning application

Parameter sharing is the method of sharing weights by all neurons in a particular feature map

ENSEMBLE METHODS:

- Ensemble methods are techniques that aim to improving the results in models by combining multiple model instead of using single model.
- The combined models increase the accuracy of the results.
- The most popular ensemble methods are bagging boosting

SENSEMBLE METHODS:

- BAGGING
- BOOSTING

BOOTSTACKING:

- The argumanted data is trained with multiple models in AI process the accuracy results is more

- Ensemble methods are ideal for regression and classification where they reduce bias and variance to accuracy of models

BAYES THEORM:

A bayes theorm finds the probability of an event occuring given the probability of another event that has already occurs is called bayes theorm.

P(A/B) = P(B/A) P(A) / P(B)

Ex: if we toss a coin the probabilites = heads and tails
the probability of getting heads = 50% The probability of getting tails = 50%
The occuring probability = 100%

We want to know that having alley when the text says
P(Y) = 1% * 80% + 99% +10%
= 10.7%

P(A/Y) = P(A) P(Y/A) / P(Y)
= 1% + 80% / 10.7%
= 7%

LSTM - LONG SHORT TERM MEMORY:

- LSTM network is a type of recurrent neural network architecture that is designed by the problem of vanishing expanding gradients in traditional RNN.
- LSTM are widely used in deeplearning for sequential data analyzing such as speech recoginization, NLP, & time series analysis
- The architecture of ISTM network is similar to that trend RNN, but it includes memory cell & three gates

 Input gates

- Forget gates
- Output gates

Restricted boltzman machine:

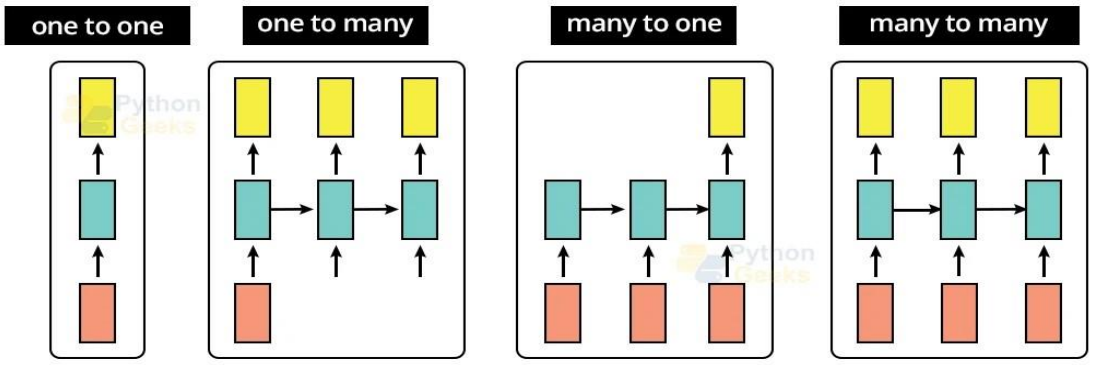
- A RBM is a type of artificial neural networks that commonly used in unsupervised machine learning tasks, such as dimensionally reduction, feature learning, and collebrative learning to represent the probability distribution
- The RBM consist of 2 layers visible layer and hidden layer
- The visible layers represents the input data and the hidden layers represents a set of latent variables that captures the under lying because they dont have common connections

RNN-- recurrent neural networks

- In RNN we have seprate and indipendent input and output layers
- Which we inefficient the dealing with sequencial data
- Hence a new neural network called rnn was introdued to store of previous outputs in the internal memory
- These results are then fed into the neural network as input this allows it to used in applications like pattern detection speech recoginization, nlp, time series predection
- RNN has hidden layers that acts memory locations to store the output of a layer in a loop

There are 4types are there in recurrent neural network

- One to one
- One to many
- Many to one
- Many to many



1.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

2.One to many:

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

3.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

4.Many to Many

- It is used to generate the sequence of output data from a sequence of input data

Auto encoders:

- An auto encoder AI 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 consists of two parts
- Encode
- Decode

Types of auto encoders:

- Vanila autoencoders
- Convolutional auto encoders
- Recurrent auto encoders
- Variational auto encoders
- Denoising auto encoders
- Adversial auto encoders

Google net architecture:

Google net:
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

SOURCE CODE

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 35 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

importing packages

In [28]:
```
import numpy as np
import pandas as pd
import pickle
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import ConfusionMatrixDisplay , classification_report ,f1_score
```

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 35 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Loding data

In [29]:
```
df = pd.read_csv("C:\\Users\\s.niranjan sai\\Downloads\\archive (3)\\creditcard_2023.csv")
df.head()
```

Out[29]:

	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	
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
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
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
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
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

5 rows × 31 columns

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 36 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Rows and columns in a dataset

In [30]:
```
df.shape
```

Out[30]: (568630, 31)

basic information

In [31]:
```
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
```

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 37 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
 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 FRAUD DETECTION TEAM-5 DS Last Checkpoint: 37 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Checking NULL values

In [32]:
```
df.isna().sum()
```

Out[32]:
```
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
```

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 38 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Statistical information about the data

In [33]:
```
df.describe()
```

Out[33]:

	id	V1	V2	V3	V4	V5	V6	V7
count	568630.000000	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05
mean	284314.500000	-5.638058e-17	-1.319545e-16	-3.518788e-17	-2.879008e-17	7.997245e-18	-3.958636e-17	-3.198898e-17
std	164149.486121	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00
min	0.000000	-3.495584e+00	-4.996657e+01	-3.183760e+00	-4.951222e+00	-9.952786e+00	-2.111111e+01	-4.351839e+00
25%	142157.250000	-5.652859e-01	-4.866777e-01	-6.492987e-01	-6.560203e-01	-2.934955e-01	-4.458712e-01	-2.835329e-01
50%	284314.500000	-9.363846e-02	-1.358939e-01	3.528579e-04	-7.376152e-02	8.108788e-02	7.871758e-02	2.333659e-01
75%	426471.750000	8.326582e-01	3.435552e-01	6.285380e-01	7.070047e-01	4.397368e-01	4.977881e-01	5.259548e-01
max	568629.000000	2.229046e+00	4.361865e+00	1.412583e+01	3.201536e+00	4.271689e+01	2.616840e+01	2.178730e+02

8 rows × 31 columns

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 38 minutes ago (autosaved) Logout

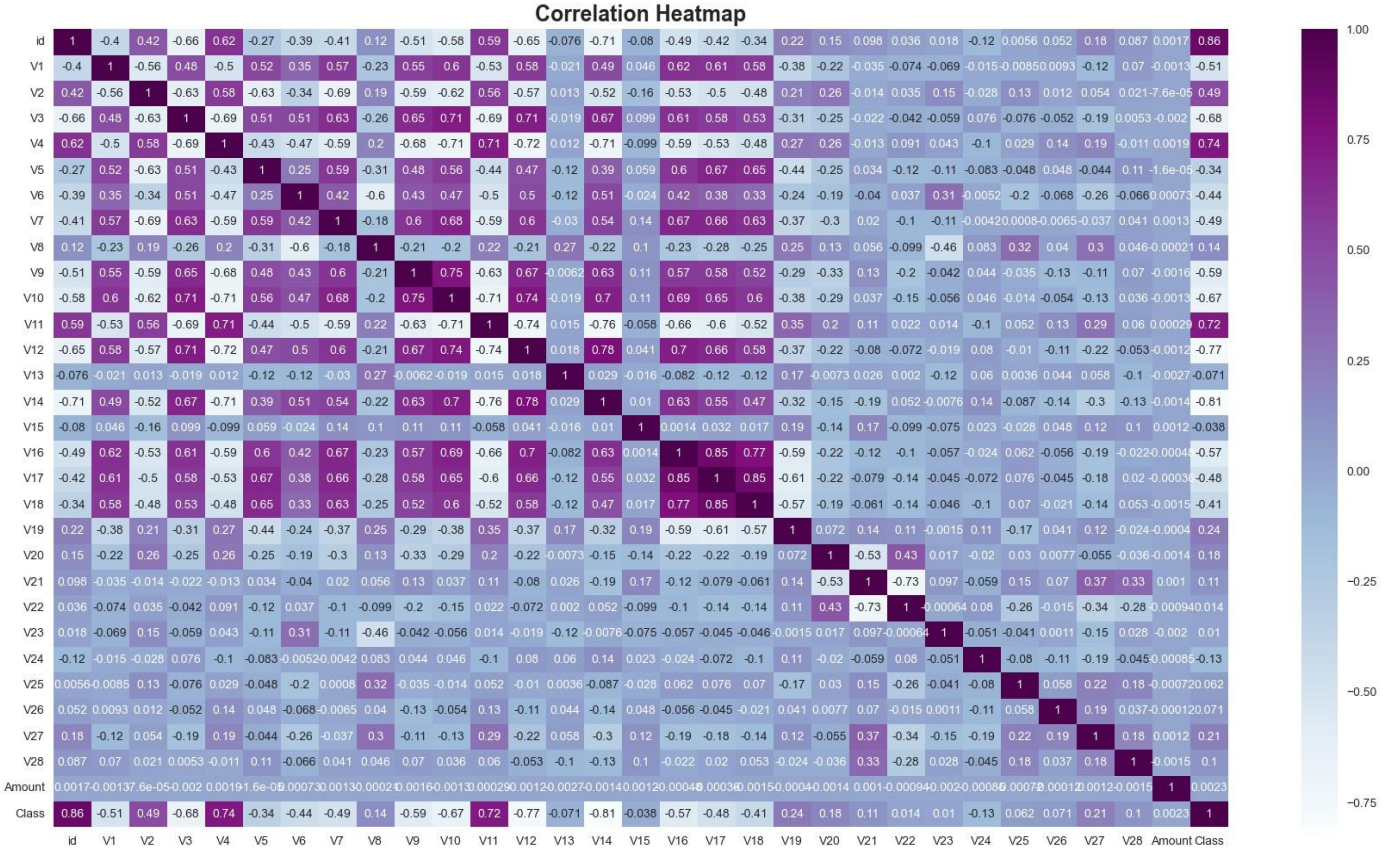
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

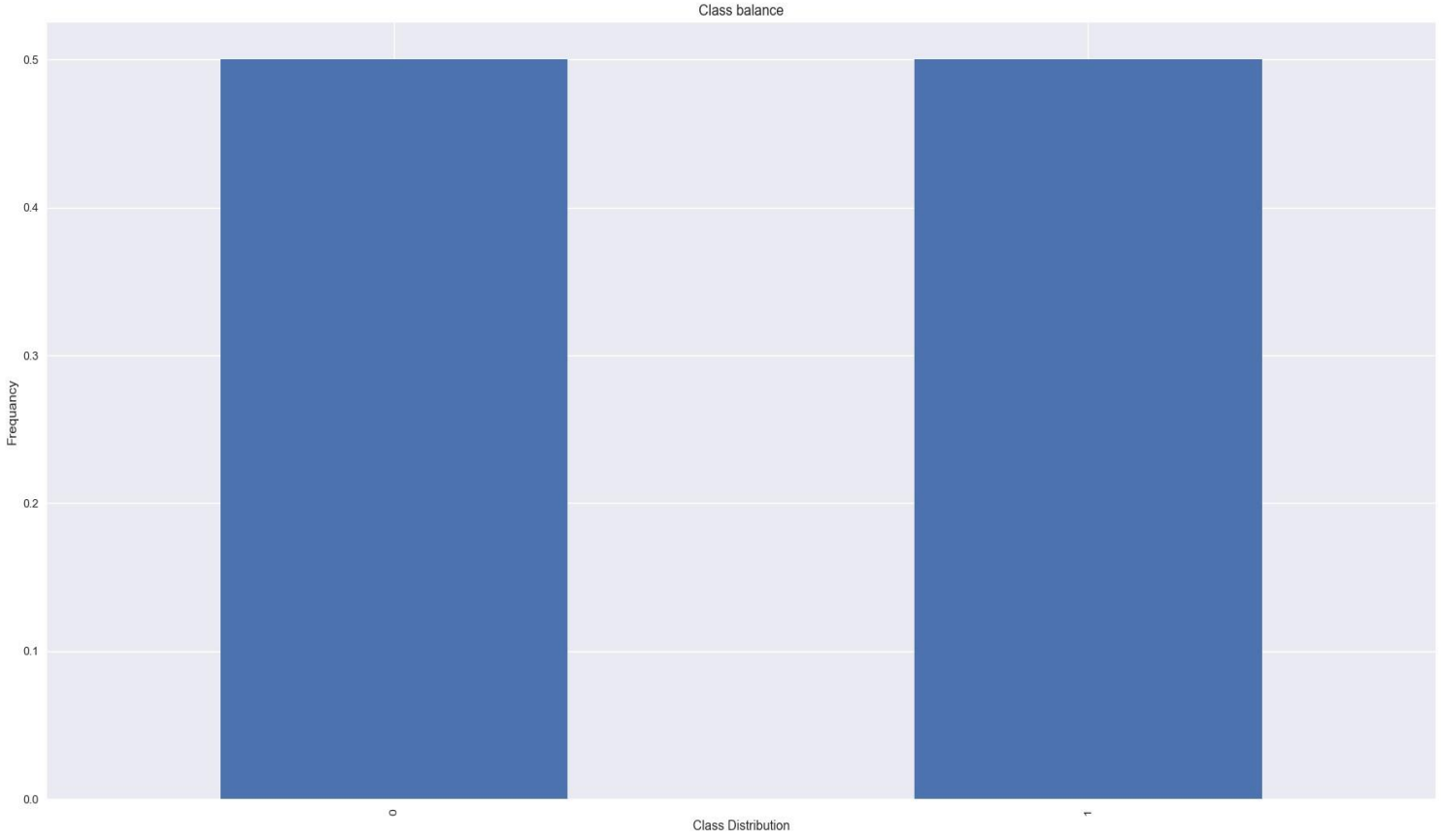
checking correlation by using Heatmap

In [34]:
```
#corr = df.drop(columns=['Class']).corr()
#sns.heatmap(corr);
plt.style.use("seaborn")

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

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

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

plt.show()
```

```
C:\Users\s.niranjan sai\AppData\Local\Temp\ipykernel_7924\3672579383.py:3: MatplotlibDeprecationWarni
ng: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond t
o the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_8-<style>'. Alter
natively, directly use the seaborn API instead.
  plt.style.use("seaborn")
```



```
In [35]: df['Class'].value_counts(normalize= True).plot(kind= 'bar')
         plt.xlabel("Class Distribution")
         plt.ylabel("Frequancy")
         plt.title("Class balance");
```



jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 40 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Training data and Testing data

In [40]:
```
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,)
```

Accuracy

In [41]:
```
acc_baseline = y_train.value_counts(normalize=True).max()
print("Baseline Accuracy:", round(acc_baseline, 4))
```

```
Baseline Accuracy: 0.5002
```

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 39 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Data cleaning

In [36]:
```
x= df.drop(['id', 'Class'], axis= 1)
y= df['Class']
```

In [37]:
```
stn_scaler = StandardScaler()

x_scaled = stn_scaler.fit_transform(x)
```

In [38]:
```
X = pd.DataFrame(x_scaled,columns=x.columns)
```

In [39]:
```
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,)
```

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 40 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [42]:
```
clf = LogisticRegression()
```

In [43]:
```
clf.fit(X_train , y_train)
```

Out[43]:
```
LogisticRegression
LogisticRegression()
```

In [44]:
```
clf.predict(X_train)
```

Out[44]:
```
array([1, 1, 1, ..., 1, 0, 0], dtype=int64)
```

In [45]:
```
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)}")
```

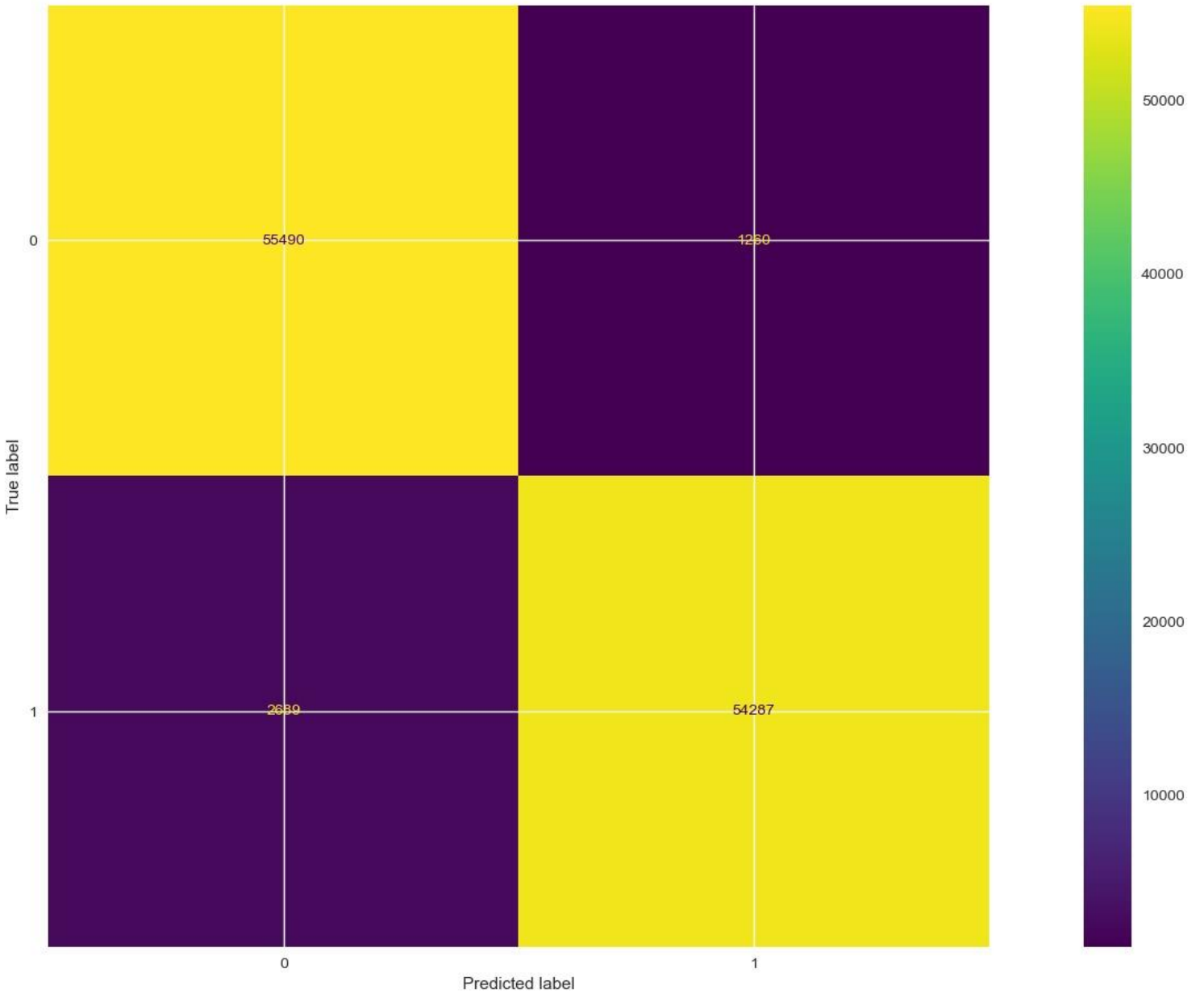
```
Training accuracy: 0.9649
test accuracy: 0.9653
```

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 41 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [46]:
```
ConfusionMatrixDisplay.from_estimator(

    clf,
    X_test,
    y_test

);
```


55490
1250
2689
54287
0
1
True label
0
1
Predicted label
50000
40000
30000
20000
10000

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 42 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [47]:
```
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
```

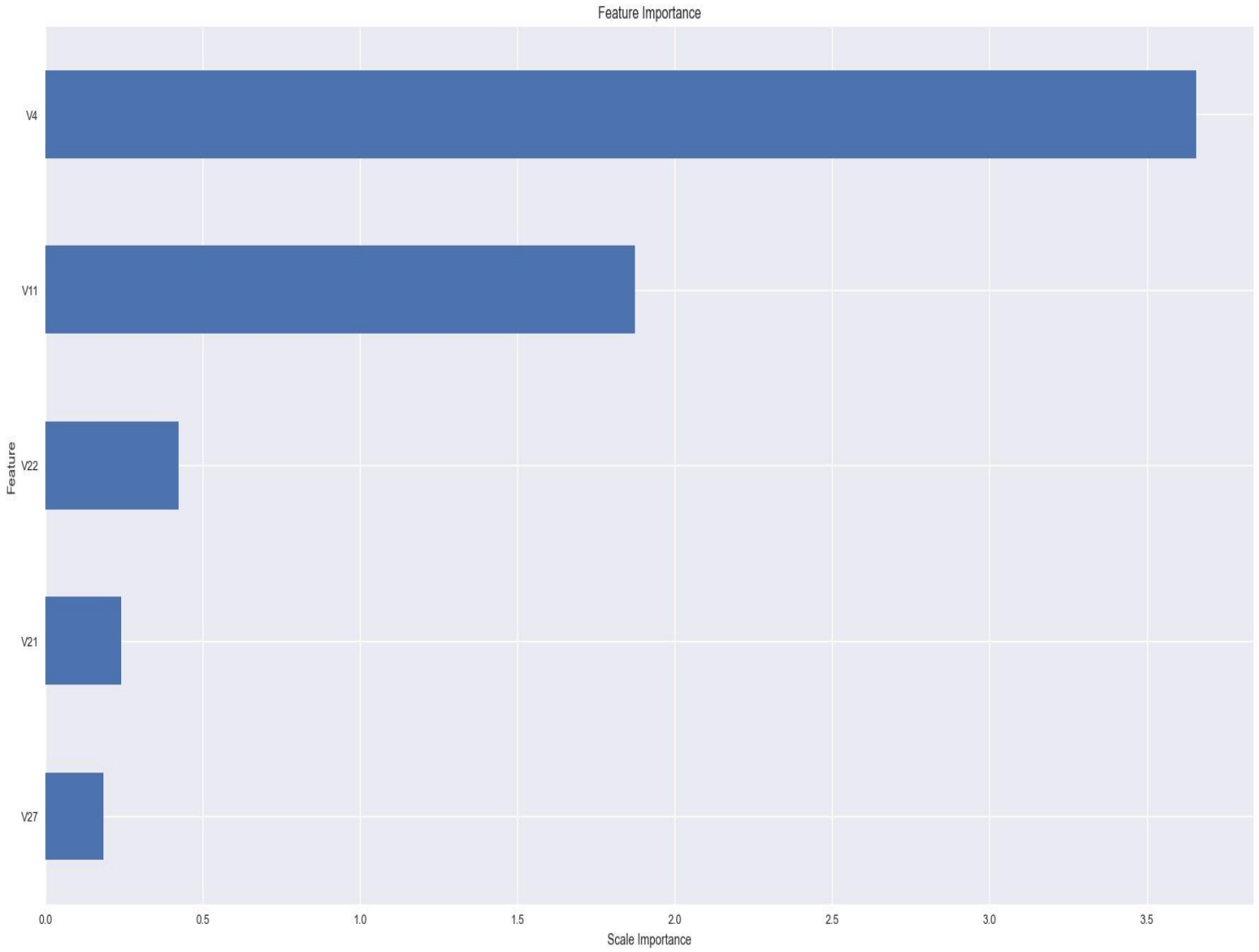
In [48]:
```
features = X_test.columns
importances = clf.coef_[0]
```

jupyter FRAUD DETECTION TEAM-5 DS Last Checkpoint: 42 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [49]:
```
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");
```


Feature Importance
V4
V11
V22
V21
V27
Feature
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Scale Importance

CONCLUSION

In conclusion, Machine Learning and Artificial Intelligence techniques implemented through Python offer a powerful and groundbreaking approach to **Credit card fraud detection**. Machine learning has significantly improved the detection of credit card fraud by enabling real-time analysis and pattern recognition in transaction data. By leveraging advanced algorithms and continuous monitoring, financial institutions can effectively mitigate fraud, protect customers, and maintain trust in their services.