# PREDICTIVE MODEL FOR GST

**1.Introduction:**

**GST:** The Goods and Services Tax (GST), introduced in India on 1 July 2017, consolidates various national and state-level taxes on goods and services. Its goal is simplify tax calculations and collection, bringing greater efficiency and transparency to India's taxation system.

**The Classifiers:**

1.)*Support Vector Classifier (SVC):*

SVC is a specific implementation of the Support Vector Machine algorithm that is designed specifically for classification tasks. In other words, SVC is an SVM used for classification. It seeks to find the hyperplane that best separates the data points into different classes

*2.)Naive Bayes:*

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable.

*3.)Artificial Neural Networks(ANN):*

high amount of output classes and high amount of data to support the performance of the model.

Neural networks are loosely representative of the human brain learning. An Artificial Neural Network consists of Neurons which in turn are responsible for creating layers. These Neurons are also known as tuned parameters.The output from each layer is passed on to the next layer

*4.)Decision Tree Classifier:*

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

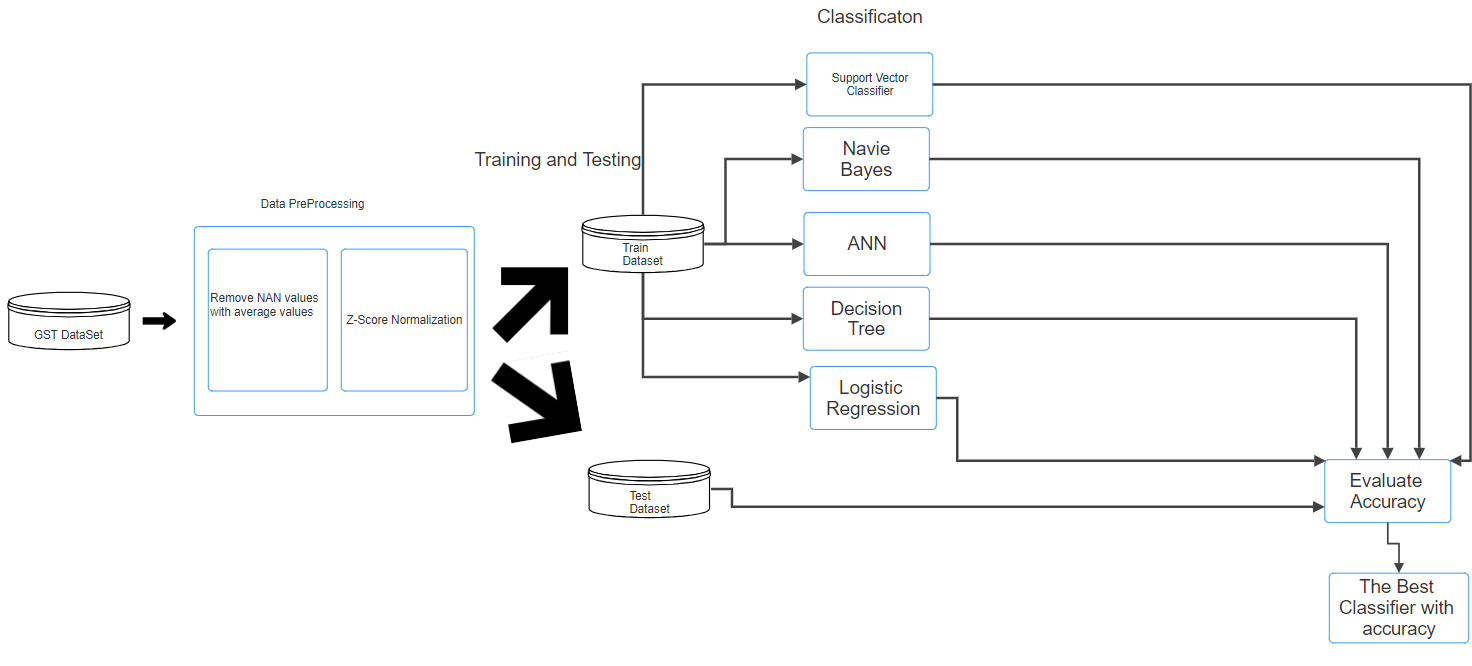
*5.)Logistic Regression:*

logistic regression machine learning is a statistical method that is used for building machine learning models where the dependent variable is used to describe data and the relationship between one dependent variable and one or more independent variables.

1. **Methodology:**

**A.)***GST Dateset-Source:*

https://innovateindia.mygov.in/online-challenge-for-developing-a-predictive-model-in-gst/



1. )Data Pre-processing:

*1.)Libraries used:*

import pandas as pd.

import numpy as np.

from sklearn.naive\_bayes import GaussianNB.

from sklearn.svm import SVC.

from sklearn.tree import DecisionTreeClassifier.

from sklearn.linear\_model import LogisticRegression.

from sklearn.model\_selection import train\_test\_split.

from sklearn.metrics import accuracy\_score, classification\_report, precision\_score, recall\_score, f1\_score, roc\_auc\_score.

from sklearn.preprocessing import StandardScaler.

import tensorflow as tf.

from tensorflow.keras.models import Sequential.

from tensorflow.keras.layers import Dense, Input.

2.)*Removing NAN values:*

*missing\_vals=["NAN","",None,"NA"]*

*X\_Train\_missing\_values=X\_Train\_Input.isin(missing\_vals)*

*X\_Train\_mean= X\_Train\_Input.mean()*

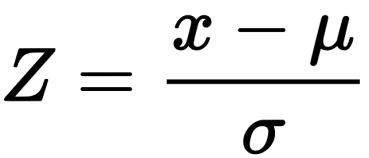
*X\_Train\_Input=X\_Train\_Input.mask(X\_Train\_missing\_values, X\_Train\_mean,axis=1)*

*#The missing values like NAN,None,etc..are replaced by mean of the values in #x\_train, similarly for x\_test as well.*

*3.)Normalization:Normalization is used to convert the values in a table/dataset into a set of values which range from 0.0 - 1.0.*

*In this we have used Z-Score Scaler.*

*Z-Score:Z-score is a statistical measurement that describes a value's relationship to the mean of a group of values. Z-score is measured in terms of standard deviations from the mean*



scaler = StandardScaler()

X\_Train\_scaled = scaler.fit\_transform(X\_Train\_Input)

X\_Test\_scaled = scaler.transform(X\_Test\_Input)

*4.)Training\_testing:*

*X\_Train, X\_Test, Y\_Train, Y\_Test = train\_test\_split(x\_test\_scaled, Y\_Test\_Input, test\_size=0.2, random\_state=42)*

*#ravel() is used to covert the Y data into 1D array from 2D array*

*#Why? == The classifier expects only 1D array*

*Y\_Train=Y\_Train.values.ravel()*

*Y\_Test=Y\_Test.values.ravel()*

*5.)Classification:*

*A.)Model Selection:*

Support Vector Classifier

Gaussian Naive Bayes

Decision Tree Classifier

Logistic Regression

Artificial Neural Network (ANN)

*B.).Training and Evaluating Models:*

#All the models are taken into a dictonary and are iterated over them except for ANN.

for model\_name, model in models.items():

model.fit(X\_Train, Y\_Train)

y\_pred\_train = model.predict(X\_Train)

y\_pred\_test = model.predict(X\_Test)

C.)*Artificial Neural Network (ANN) Implementation:*

*a.)Building the Model:*

#To build the model in a layered manner

model = Sequential()

model.add(Input(shape=(22,)))

model.add(Dense(12, activation='relu'))

model.add(Dense(8, activation='relu'))

model.add(Dense(3, activation='softmax'))

*b).Compiling and Training:*

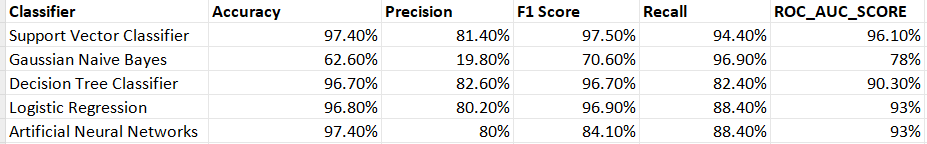
*#*sparse\_categorical\_crossentropy = It is used to compute the cross-entropy loss #between the predicted probabilities and the true class labels.

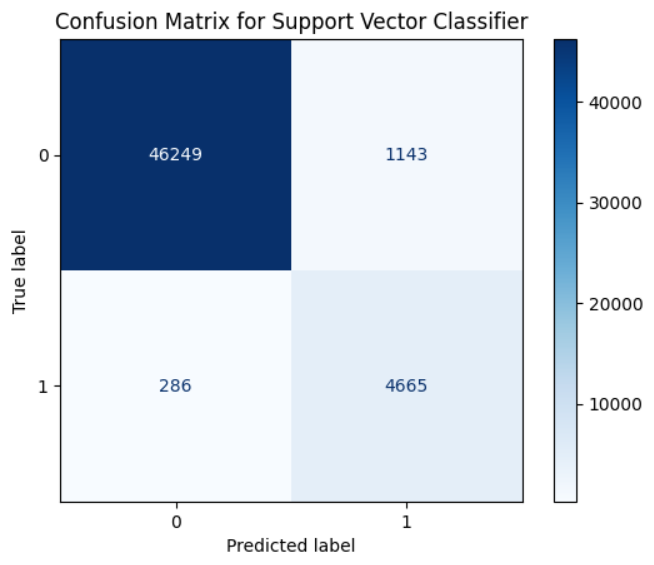
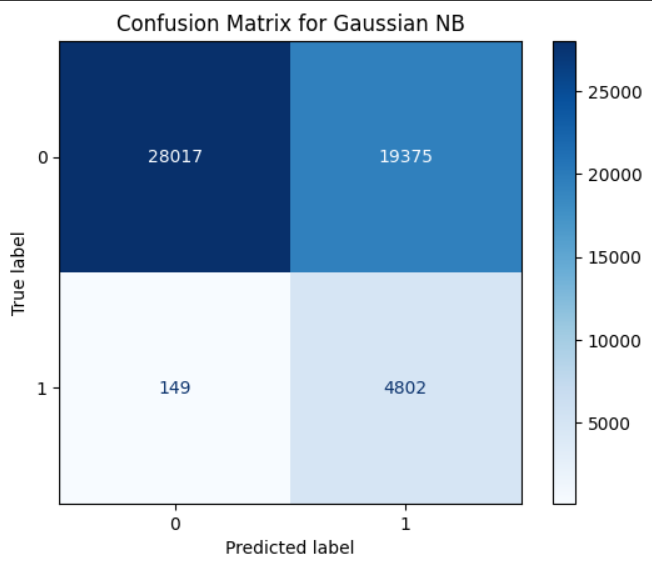
#adam= update the model's weights during training to decrease the loss.

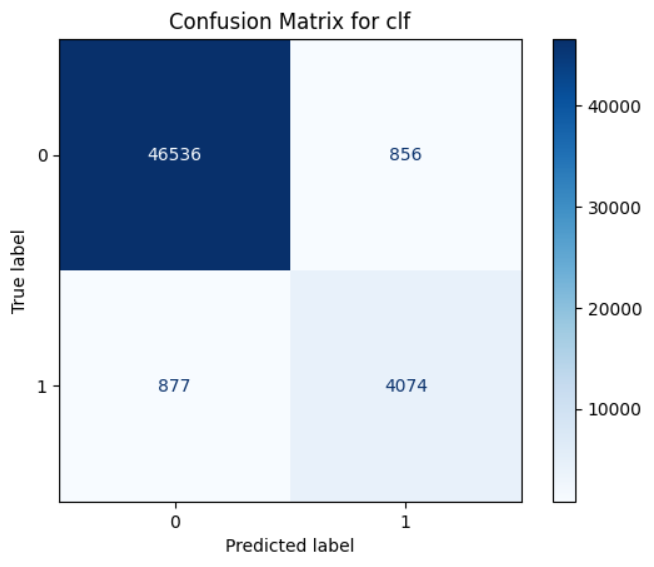
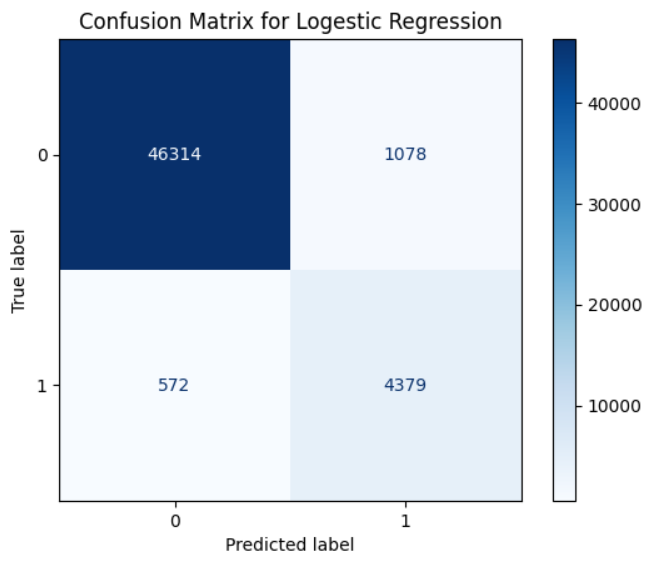
model.compile(loss='sparse\_categorical\_crossentropy', optimizer='adam', metrics=['accuracy']).

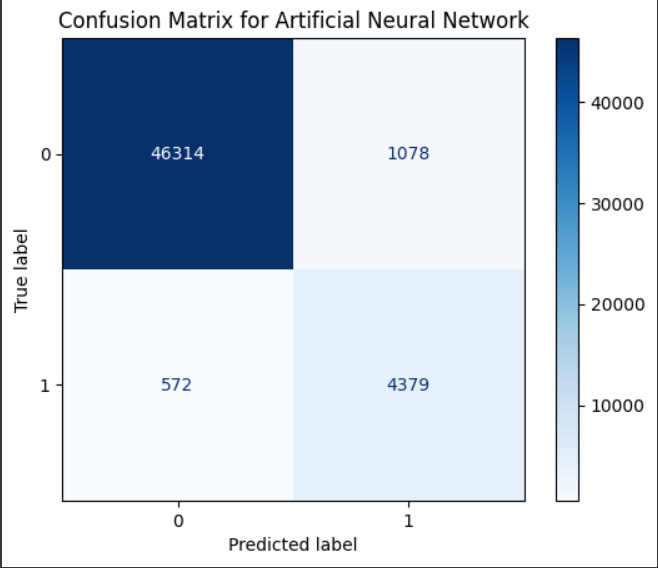
model.fit(X\_Train, Y\_Train, epochs=150, batch\_size=150, verbose=0).

**3.)Result:**



**

**4.)Conclusion:**

The best classifier for GST Data-Set is Support Vector Classifier,having accuracy=97%.

The performance of the classifiers can be increased by using Ada boost.

Team:GSTN\_856