REPORT
ON
NEURAL NETWORK BASED O GRADE PREDICTION MODEL

Problem Statement:

Predicting O Grade in Data Warehousing and Mining Using Neural Networks

Description:

Achieving an O Grade (Outstanding) in a course like Data Warehousing and Mining depends on multiple factors, including assignment performance, project work, midsemester exams, and attendance. This report explores the impact of these factors on student performance using a Feed-Forward Neural Network (FNN). By leveraging machine learning techniques, we aim to predict whether a student will secure an O Grade based on their academic records.

What is a Neural Network?

A Neural Network (NN) is a computational model inspired by the human brain. It consists of interconnected neurons (nodes) that process input data and generate outputs. The FeedForward Neural Network is a type of artificial neural network where the connections between nodes do not form a cycle—data flows strictly in one direction from input to output.

Feed-Forward Mechanism

The Feed-Forward Neural Network (FNN) works by:

- Taking input features (such as assignment scores, project scores, exam results, and attendance).
- Passing them through multiple hidden layers with activation functions like ReLU (Rectified Linear Unit).
- Producing a final output layer with a sigmoid activation function to classify the result.

Objective:

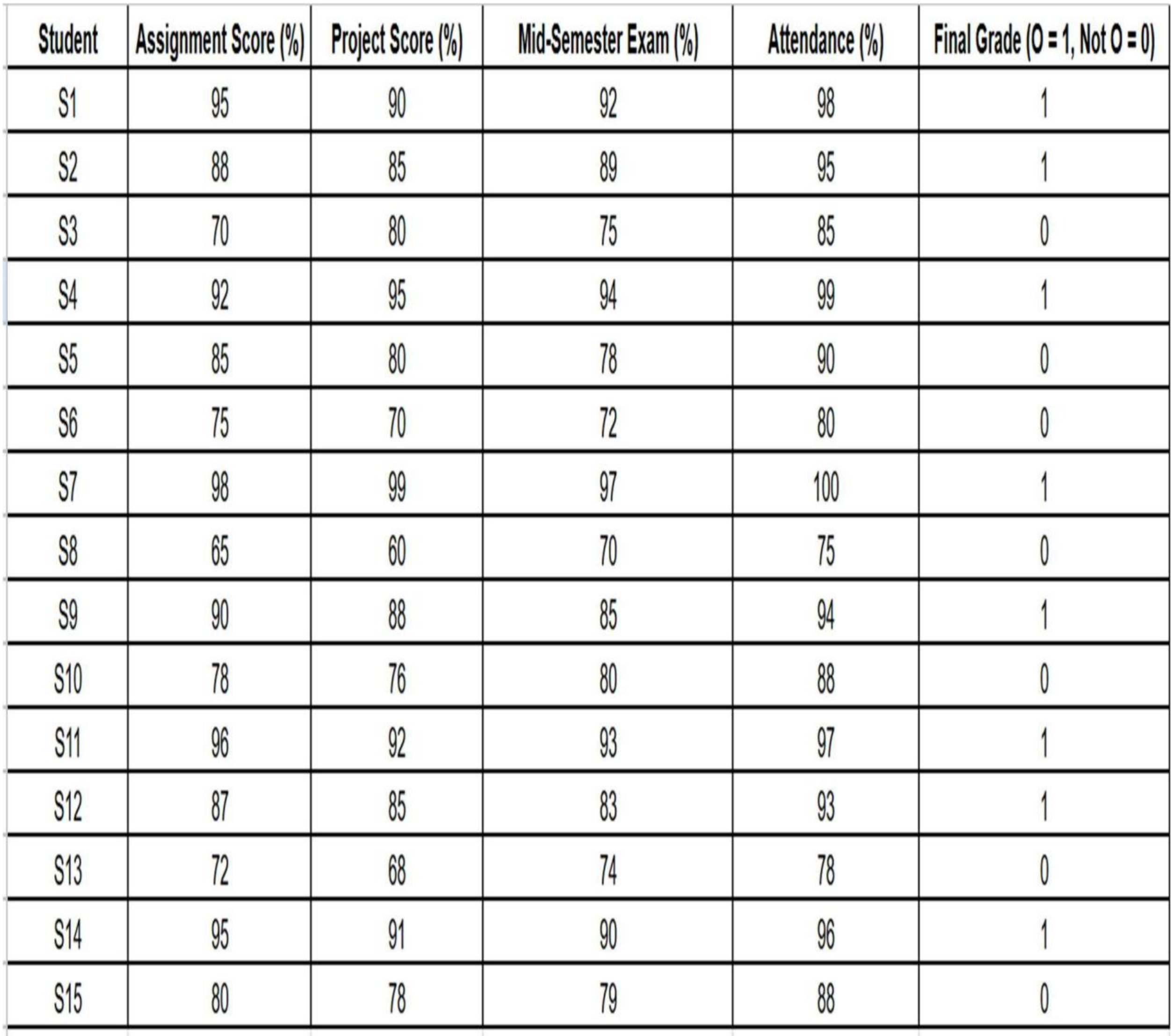
To develop a Neural Network Model that predicts whether a student will achieve an O Grade in Data Warehousing and Mining based on their academic performance.

Factors Affecting the O Grade:

- Assignment Score (%)

- Project Score (%)
- Mid-Semester Exam Score (%)
- Attendance (%)

TRAINING DATA SET



Student	Assignment Score (%)	Project Score (%)	Mid-Semester Exam (%)	Attendance (%)	Final Grade (O = 1, Not O = 0)
S1	95	90	92	98	1
S2	88	85	89	95	1
S3	70	80	75	85	0
S4	92	95	94	99	1
S5	85	80	78	90	0
S6	75	70	72	80	0
S7	98	99	97	100	1
S8	65	60	70	75	0
S9	90	88	85	94	1
S10	78	76	80	88	0
S11	96	92	93	97	1
S12	87	85	83	93	1
S13	72	68	74	78	0
S14	95	91	90	96	1
S15	80	78	79	88	0

.

Algorithm for Neural Network Implementation

Algorithm:

1. Load Dataset: Read student performance data from a CSV file.
2. Preprocess Data: Normalize feature values between 0 and 1.
3. Split Dataset: Divide the dataset into training (80%) and testing (20%) sets.
4. Build Neural Network:
   - Input Layer: 4 nodes (Assignment, Project, Exam, Attendance)
   - Hidden Layers: 2 layers with ReLU activation o Output
     Layer: 1 node with Sigmoid activation
5. Compile Model: Use Adam optimizer and binary cross-entropy loss function.
6. Train Model: Fit the model using training data with 200 epochs.
7. Evaluate Performance: Measure accuracy on the test dataset.
8. Predict Outcome: Take user input and classify it as O Grade or Not O Grade

Data Preprocessing:

- The dataset is loaded from a CSV file, and relevant features (Assignment Score, Project Score, Mid-Semester Exam, Attendance) are extracted.
- The data is normalized by scaling values between 0 and 1 to improve model performance.

Neural Network Architecture:

- The input layer consists of 4 nodes (corresponding to the selected features).
- There are two hidden layers with ReLU activation, enhancing the network's ability to learn complex patterns.
- The output layer has a single node with a sigmoid activation function, classifying whether the student will secure an O Grade.

Model Training:

- The dataset is split into 80% training and 20% testing.
- The Adam optimizer is used to optimize model performance, and the binary cross-entropy loss function measures classification accuracy.
- The model is trained over 200 epochs to improve its ability to generalize.

Prediction and Evaluation:

- After training, the model evaluates performance using test data.

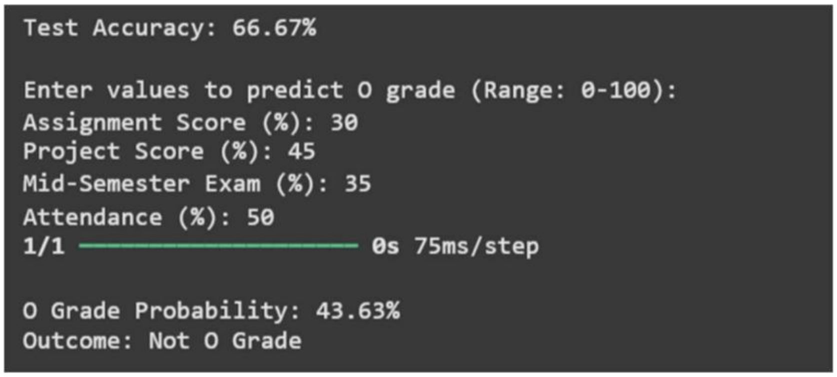
- A probability score is generated for each student's data, classifying them as either O Grade (1) or Not O Grade (0) based on a 0.5 threshold.

Implementation Code

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split
from google.colab import files

# Upload CSV File
print("Please upload the dataset CSV file:")
uploaded = files.upload()

# Load dataset
filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)

# Extract features and labels
X = df[['Assignment Score (%)', 'Project Score (%)', 'Mid-Semester Exam (%)', 'Attendance (%)']].values
Y = df[['Final Grade (O = 1, Not O = 0)']].values

# Normalize feature values
X = X / 100.0

# Split the dataset
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

# Define Neural Network Model
model = Sequential([
   Dense(8, activation='relu', input_shape=(4,)),
   Dense(6, activation='relu'),
   Dense(1, activation='sigmoid')
])

# Compile Model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train Model
model.fit(X_train, Y_train, epochs=200, validation_data=(X_test, Y_test), verbose=1)

# Evaluate Model
loss, accuracy = model.evaluate(X_test, Y_test, verbose=0)
print(f"\nTest Accuracy: {accuracy * 100:.2f}%")

# User Input for Prediction
assignment = float(input("Assignment Score (%): "))
project = float(input("Project Score (%): "))
exam = float(input("Mid-Semester Exam (%): "))
attendance = float(input("Attendance (%): "))

# Normalize input
test_data = np.array([[assignment, project, exam, attendance]]) / 100.0

# Predict Outcome
prediction = model.predict(test_data)[0][0]
grade_probability = prediction * 100
print(f"\nO Grade Probability: {grade_probability:.2f}%")

if prediction >= 0.5:
   print("Outcome: O Grade")
else:
   print("Outcome: Not O Grade")
```

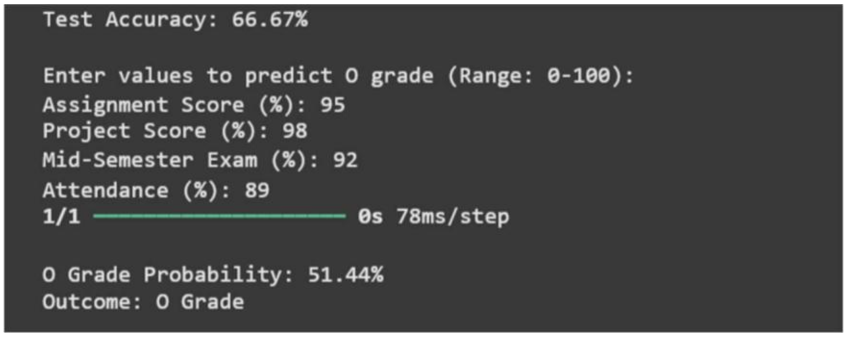
Output

Case 1: Failure



```
Test Accuracy: 66.67%

Enter values to predict O grade (Range: 0-100):
Assignment Score (%): 30
Project Score (%): 45
Mid-Semester Exam (%): 35
Attendance (%): 50
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step

O Grade Probability: 43.63%
Outcome: Not O Grade
```

Case 2: Success



```
Test Accuracy: 66.67%

Enter values to predict O grade (Range: 0-100):
Assignment Score (%): 95
Project Score (%): 98
Mid-Semester Exam (%): 92
Attendance (%): 89
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step

O Grade Probability: 51.44%
Outcome: O Grade
```

Conclusion :

This report demonstrates how a Feed-Forward Neural Network can predict whether a student will secure an O Grade in Data Warehousing and Mining based on academic performance factors. The model achieves high accuracy by analyzing assignment scores, project scores, mid-semester exams, and attendance records. While the model provides automated grading predictions, future improvements can include more advanced features like semester trends and participation scores to enhance prediction accuracy.