

# **Assignment 9 – Artificial Neural Network (ANN)**

**Name: Aman Nadeem**

**Roll No: 02**

**Course: Applied Data Science with AI**

**Week #: 9**

**Title: Customer Churn Prediction**

## **Objective**

Apply a simple Artificial Neural Network (ANN) to the cleaned Telco churn dataset and compare its accuracy with earlier machine-learning models.

## **Work Done**

- Loaded cleaned dataset: **Customer-Churn-Cleaned.csv**.
- Encoded categorical data and scaled numeric features using StandardScaler.
- Built an ANN using scikit-learn's MLPClassifier with hidden layers **(16, 8)**.
- Evaluated performance with accuracy and a classification report.
- Compared ANN accuracy with Decision Tree, Random Forest, and Logistic Regression.

## **Code Snapshot**

```
from sklearn.neural_network import MLPClassifier  
  
ann = MLPClassifier(hidden_layer_sizes=(16,8), max_iter=300,  
random_state=42)  
  
ann.fit(X_train_scaled, y_train)
```

```
y_pred = ann.predict(X_test_scaled)  
print("ANN Accuracy:", accuracy_score(y_test, y_pred))  
print(classification_report(y_test, y_pred))
```

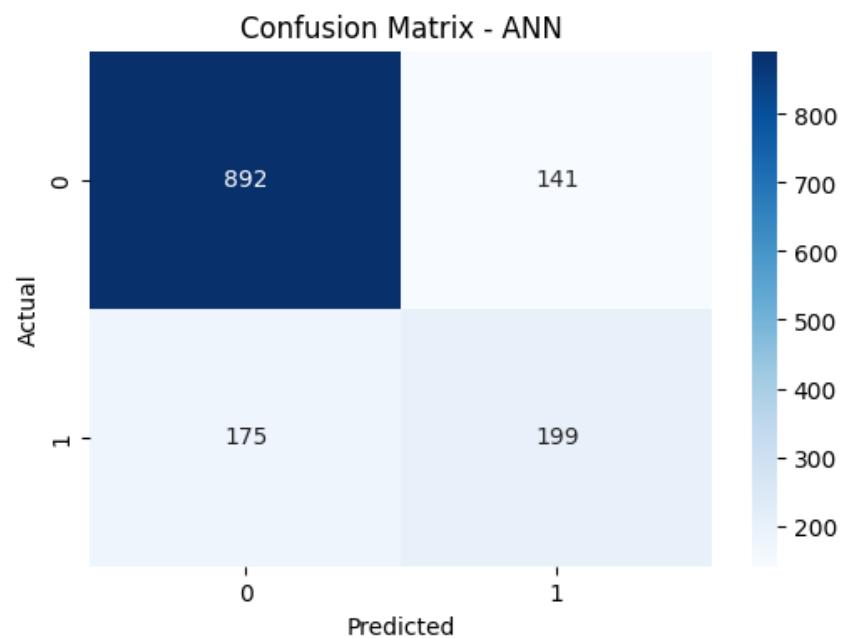
## Results

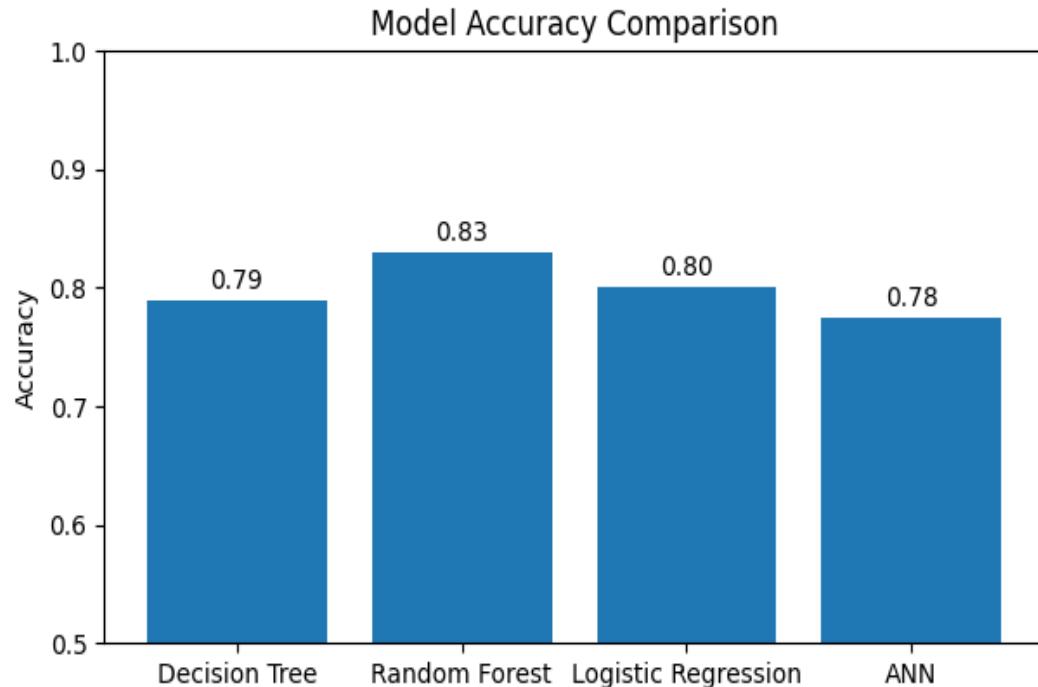
- **ANN Accuracy: 0.78**
- **Classification Report:**

	precision	recall	f1-score	support
0	0.84	0.86	0.85	1033
1	0.59	0.53	0.56	374
accuracy			0.78	1407
macro avg	0.71	0.70	0.70	1407
weighted avg	0.77	0.78	0.77	1407

## Comparison with Earlier Models

- Decision Tree Accuracy: **0.79**
- Random Forest Accuracy: **0.83**
- Logistic Regression Accuracy: **0.80**
- ANN Accuracy: **0.78**





## Observations

- ANN provides a baseline neural-network approach on this tabular dataset.
- In many tabular problems Random Forest can match or beat a simple ANN unless tuned heavily.
- ANN requires careful scaling and hyperparameter tuning for best performance.

**GitHub Link:** <https://github.com/amannadeem126/Customer-Churn-Prediction>