**BUILDING AND TRAINING A MULTILAYER PERCEPTRON (MLP) FOR CLASSIFICATION**

**Table of Contents**

[1. Introduction 4](#_Toc184917204)

[2. Dataset Overview 4](#_Toc184917205)

[Key Features: 4](#_Toc184917206)

[3. Preprocessing 5](#_Toc184917207)

[3.1. Data Cleaning 5](#_Toc184917208)

[3.2. Encoding Categorical Variables 5](#_Toc184917209)

[3.3. Scaling Numerical Features 5](#_Toc184917210)

[3.4. Splitting the Dataset 5](#_Toc184917211)

[3.5. One-Hot Encoding the Target Variable 6](#_Toc184917212)

[4. Building the MLP Model 6](#_Toc184917213)

[4.1. Model Architecture 6](#_Toc184917214)

[4.2. Model Compilation 6](#_Toc184917215)

[4.3. Training the Model 7](#_Toc184917216)

[5. Evaluating the Model 7](#_Toc184917217)

[6. Visualizing Model Performance 7](#_Toc184917218)

[6.1. Accuracy Plot 7](#_Toc184917219)

[6.2. Loss Plot 8](#_Toc184917220)

[7. Challenges and Considerations 8](#_Toc184917221)

[7.1. Overfitting 8](#_Toc184917222)

[7.2. Imbalanced Classes 8](#_Toc184917223)

[7.3. Feature Importance 9](#_Toc184917224)

[8. Applications of MLP 9](#_Toc184917225)

[9. Future Improvements 9](#_Toc184917226)

[10. Conclusion 10](#_Toc184917227)

[Reference 11](#_Toc184917228)

[Appendix 13](#_Toc184917229)

# 1. Introduction

Artificial Neural Network is one of the categories of Artificial Intelligence which has remained very useful in locating possible solutions to complicated issues in different fields. These includes; one amongst the most popular and fundamental types of supervised learning is the Multilayer Perceptron (MLP). An MLP is defined by the fact that it can solve nonlinear problems by having multiple layers, an input layer, one or more hidden layers, and an output layer. All layers include neurons which perform inputs processing with help of weights, bias and activations. Here, we explain the general approach to constructing and training an MLP and apply the knowledge to a binary classification problem with the customer dataset to predict membership churn.

The dataset contains plenty of customer details embracing demography details, utilization of services, and billing history. The is to provide an estimate of the likelihood of a customer quitting the membership program that is called “churn”, so that organizations may put strategies into place. This tutorial takes the student through the preprocessing, model architecture, training, and assessing steps required in constructing an MLP using Python and well-known packages such as TensorFlow, Scikit-learn, and Matplotlib.

# 2. Dataset Overview

The data set chosen for this tutorial is centered on customer churn prediction. There are several categorical and numeric predictors, and a single binary outcome variable called LeftMembership, reflecting a customer’s churn or non-churn status. The features include:

## Key Features:

* **Customer Demographics:** Gender, older-generation customer, and whether they have dependents.
* **Service Details:** This was achieved depending on the classification of the services into Internet Service type, Phone Service, and additional services such as Streaming TV or Movies.
* **Contract Information:** Paperless billing, and the duration of the contract the contract type will include Month-to-Month, One-Year, etc.
* **Billing Details:** Monthly charges and total charges also showed significant improvement **when comparing them with the previous fiscal year.**
* **Target Variable:** LeftMembership that varies in the binary set having 0 as non-churn and 1 as churn.

Since the features include categorical variables and numerical variables, the preprocessing of data is critical step prior to MLP model building.

# 3. Preprocessing

## 3.1. Data Cleaning

The dataset involves some columns, for instance, the CustomerID which are not actually required for the prediction process. These columns are eliminated in order to reduce the data points. The handling of missing values in the reproductive health data set is done to eliminate biased results that may be occasioned by incorporation of missing or null values into the training process.

## 3.2. Encoding Categorical Variables

Character attributes like Gender, Internet Service, and Contract has categorical features and as such will be encoded using LabelEncoding which labels every feature depending on its value. For instance, the Gender, which is initially presented as text data, can be one-hot encoded as: 0 for Male and 1 for Female data type. In the same way, other categorical features are also encoded to sit well for the numerical computations that the MLP model will require.

## 3.3. Scaling Numerical Features

As for the field data, some of the features such as Monthly Charges and Total Charges have a different range of scale which will affect the performance of the MLP model when used. To deal with this, what is referred as Standard Scaling is used so as to make the values’ mean to be equal to 0 and the standard deviation equal to 1. This means all features pay their way in terms of model training through the contribution proportionate to their value.

## 3.4. Splitting the Dataset

The dataset features a training set as well as the testing set, so the performance is going to be tested on unseen data. An important part of the experiments is a stratified split to preserve the distribution of classes in the target variable.

## 3.5. One-Hot Encoding the Target Variable

Thus, a binary class variable, called LeftMembership, is one-hot encoded for employing TensorFlow’s categorical cross entropy as the loss function.

# 4. Building the MLP Model

An MLP is made up of a neural network of layers which are interconnected. In this tutorial, we construct an MLP using TensorFlow’s Keras environment that can make the process of creating and training a neural network much easier.

## 4.1. Model Architecture

The model architecture includes:

1. **Input Layer:** Takes the feature vector that the the dataset has preprocessed.
2. **Hidden Layers:** Two more new layers are introduced and both of them are hidden layers that contain a specific number of neurons, 64 neurons and 32 neurons respectively of ReLU activation function introduced to incorporate non-linearity.
3. **Dropout Layers:** Standard Dropout: The model is used to cut out or reduce signal propagation to a fraction of neurons during the training phases in order to prevent cases of overfitting.
4. **Output Layer:** A softmax activation function is applied to generate probabilities of two classes, that is, churn and non-churn.

## 4.2. Model Compilation

The model is compiled using:

* **Optimizer:** Adam, that changes the learning rate during the training process for improving convergence.
* **Loss Function:** Categorical Cross-Entropy is the best for the multi-class classification problem.
* **Metrics:** Calibration, to measure the accuracy of a model.

## 4.3. Training the Model

The model is trained using the fit method, with:

* **Epochs:** Namely, the number of iterations required to run through the entire training data sets.
* **Batch Size:** The number of samples for which the model had to predict before its weights were updated.
* **Validation Split:** The training data contains a sub-set of validation data to check the performance of the model at run-time training phase.

# 5. Evaluating the Model

After such training the evaluate method is used to test the performance of the model on the test set. Key metrics include:

* **Loss:** Shows goodness of fit of the model on the data.
* **Accuracy:** Indicates the fraction of the all samples fitted into the correct classes.

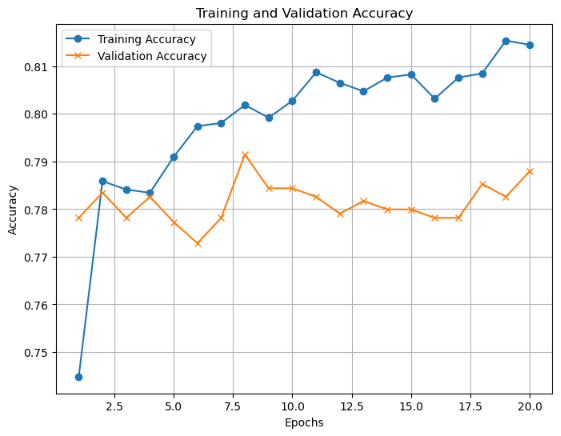
The accuracy of the test gives us a better indication of the real world performance of the model.

# 6. Visualizing Model Performance

## 6.1. Accuracy Plot

To visualise the training and validation accuracy, an accuracy plot is created as shown below: This is useful in connecting overfitting or underfitting Out of all of the methods for evaluating a model’s performance, this seems to be the most beneficial in linking overfitting or underfitting. The plot typically shows:

* **Training Accuracy:** How well a given model does on its training dataset.
* **Validation Accuracy:** How well the created model would generalize on unseen data.



#### Fig 1: Accuracy Plot

(Source: Self-Made)

## 6.2. Loss Plot

In the same way, a loss plot is constructed in order to monitor training and validation loss as the training process proceeds.

# 7. Challenges and Considerations

## 7.1. Overfitting

In essence, overfitting is the situation where a model has high accuracy when the program is run on the training data but low accuracy when run on other data sets. Dropout layers and the techniques of early stopping can be used to overcome this problem.

## 7.2. Imbalanced Classes

The simpler models may perform well but they are not fair in terms of the target variable since the target variable may be imbalanced to an extreme for instance CHURN may be sampled more than NON\_CHURN. There is a solution to this, for example, by methods such as class weighting, or SMOTE (Synthetic Minority Oversampling Technique).

## 7.3. Feature Importance

Of equal importance is to know which features drive the model most. A feature importance can be assessed by such concepts as SHAP (SHapley Additive exPlanations).

# 8. Applications of MLP

MLPs are versatile and can be applied to various domains, such as:

* **Customer Retention:** Churn rate forecast in retail, telecommunication, and subscription-based industries.
* **Healthcare:** Identification of diseases based on aspects of the affected patient.
* **Finance:** Fraud detection and credit scoring are the group of applications of big data analytics for the banking sector.

These ideas of this tutorial can be applied on other similar issues across other industries.

# 9. Future Improvements

While the current MLP model achieves reasonable performance, several enhancements can be explored:

1. **Hyperparameter Tuning:** Tune up the learning rate, the batch size, the number of neurons, using simple meta-parameters search techniques as the Grid Search, the Bayesian Optimization.
2. **Ensemble Methods:** Rather than using the output of a single model, use several models to come up with a solution.
3. **Transfer Learning:** Deep learning models on a small amount of labeled data.
4. **Explainability:** Perform interpretability techniques in order to understand explanations of the model’s decisions.

# 10. Conclusion

In this tutorial, the preprocessing of data, construction, training and assessing of the MLP model for a binary classification was shown. It allowed for prediction of churn and the information given in the dataset provided a good understanding of customers’ behavior. That way, we ensured that the workflow was quite stable and amenable to optimization for a wide range of machine learning applications.

Key takeaways include:

* Why preprocessing is crucial in the case of categorical and numerical data.
* The great ability and versatility of MLPs in solving nonlinear problems.
* The place of visualization in performance analysis.

Further work can concern about fine-tuning of the model and about search for other architecture to get better performance.

# Reference

Al Bataineh, A., Kaur, D. and Jalali, S.M.J., 2022. Multi-layer perceptron training optimization using nature inspired computing. IEEE Access, 10, pp.36963-36977.

Desai, M. and Shah, M., 2021. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). Clinical eHealth, 4, pp.1-11.

Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J. and Atkinson, P.M., 2018. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS Journal of Photogrammetry and Remote Sensing, 140, pp.133-144.

Botalb, A., Moinuddin, M., Al-Saggaf, U.M. and Ali, S.S., 2018, August. Contrasting convolutional neural network (CNN) with multi-layer perceptron (MLP) for big data analysis. In 2018 International conference on intelligent and advanced system (ICIAS) (pp. 1-5). IEEE.

Botalb, A., Moinuddin, M., Al-Saggaf, U.M. and Ali, S.S., 2018, August. Contrasting convolutional neural network (CNN) with multi-layer perceptron (MLP) for big data analysis. In 2018 International conference on intelligent and advanced system (ICIAS) (pp. 1-5). IEEE.

Gaikwad, N.B., Tiwari, V., Keskar, A. and Shivaprakash, N.C., 2019. Efficient FPGA implementation of multilayer perceptron for real-time human activity classification. Ieee Access, 7, pp.26696-26706.

Sumsion, G.R., Bradshaw, M.S., Hill, K.T., Pinto, L.D. and Piccolo, S.R., 2019. Remote sensing tree classification with a multilayer perceptron. PeerJ, 7, p.e6101.

Fath, A.H., Madanifar, F. and Abbasi, M., 2020. Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems. Petroleum, 6(1), pp.80-91.

Harirchian, E., Lahmer, T. and Rasulzade, S., 2020. Earthquake hazard safety assessment of existing buildings using optimized multi-layer perceptron neural network. Energies, 13(8), p.2060.

# Appendix

