ANN PRACTICAL IMPLEMENTATION

- Read the dataframe – import and read the data.

- Divide into X and Y variable – divide the data into dependent (Y) and independent (X) features. Y is also known as the target feature. So, X will have all the columns except the target. And Y will have only the target column.

- Train and Test Splits – Test and Test split is performed to divide the data into training and testing sets. Training data is used to train the model, and the testing data is used to validate the results. In this problem statement, I have divided/split the training data two times and extracted the validation data from the training data itself such that I can use the test data to make predictions. So, 20% of the entire dataset is for test (predictions), and 80% of the training data is again split into 20% of validation and rest for actual training.

Data Preprocessing:

- Feature Engineering: We need to change some of the categorical columns to numerical for the model to understand them. In the problem statement, columns, sex, smoker and region are categorical. Columns sex and smoker can be encoded with Label Encoder since there are only two categories. For region with multiple categories, I have encoded using One Hot Encoder for simple calculations and maintain non ordinality among the categories.

  The train data has been fit-transformed (learn and apply) but the validation data needs to be just transformed (apply what is learnt).

- Drop the region column – Columns sex and smoker has been encoded inplace. But we need to drop column region to add the one hot encoded column.

- Concat one hot encoded column – The one hot encoded column needs to be concated to the train and validation datasets respectively. Before concating, the index needs to be reset in both the dataframes to avoid data mismatch.

- Scaling – Scaling is the process to scale the values of each column within a specific range to avoid bias towards certain features. I have used the Standard Scalar for scaling. Standard Scalar works on Z – Score Normalization where the mean of the data is equal to 0 and standard deviation is 1. Scaling is usually done per column (feature), with each feature being transformed independently based on its own distribution.

- Save in pickle file – The encoders and scalar needed to be saved in pickle file for its reusability during prediction. The training memory is sustained in the models saved in pickle file and is stable. Pickle file accepts input in bytes, so it is written in 'wb'.

Hyperparameter tuning and Training:

- Hyperparameter tuning refers to the process of optimizing the hyperparameters of a deep learning model to improve its performance. Hyperparameters are the parameters set before training the model and control the model's behavior and training process. Unlike model parameters (which are learned during training), hyperparameters need to be manually selected like Batch size, No of Neurons, Optimizers, Layers.

- In this project:

1) I have set the hyperparameters as: Number of hidden layers, number of neurons and optimizer.
2) Created a function, build_model, to check all the best possible hyperparameters for the model.
3) The Hyperband algorithm will help to search the best values for the hyperparameters, and the model will be built on these values.
4) Early stopping will help reduce computation time by calculating the validation loss (in case of regression). If the validation loss is gradually decreasing and at some points after it has reached the bottom it goes up, the epoch will run 3 more times (patience = 3) before stopping the epochs.
5) An epoch refers to one complete pass through the entire training dataset by the model during training. The dataset is often divided into smaller chunks called batches (or minibatches). Each batch is fed into the model sequentially during training. For each batch, the model performs a forward pass (prediction) and calculates the loss. Then, it performs a backward pass (backpropagation) to adjust the model's parameters. After all batches in the dataset have been processed, one epoch is completed.
6) The model is fitted/trained with the best hyperparameters.
7) Lastly, the accuracy score is evaluated which is the performance of the model. Here the accuracy is around 95% on the validation data.

The model is saved in .h5 format. The .h5 format, short for **Hierarchical Data Format version 5**, is the default format used by **Keras** (and TensorFlow) for saving model architectures, weights, and training configurations. Here's why .h5 is preferred for saving machine learning models.

<u>Prediction:</u>

- All the encoder, scalar and model files are downloaded in the prediction workspace.
- The independent variables are encoded and scaled, and the prediction is made.
- Final dataframe has the original Y and the predicted Y.