Assignment 3.1: ANN for Medical Diagnosis

By

*Kesini Kandasamy*

Contents

[Scenario and Instructions 2](#_Toc205548328)

[Assignment & Solutions 3](#_Toc205548329)

[Task 1: Utilise Libraries/Dataset 3](#_Toc205548330)

[Task 2: Generate at least three EDA visualisations 4](#_Toc205548331)

[Task 3: Analyse data quality 8](#_Toc205548332)

[Task 4: Prepare Features and Target Variable 9](#_Toc205548333)

[Task 5: Class Imbalance and Dataset Splitting 9](#_Toc205548334)

[Task 6: Design an Appropriate DCNN-based Architecture 9](#_Toc205548335)

[Task 7: Configure the Training Process 10](#_Toc205548336)

[Task 8: Train the model with Appropriate Batch Size and Number of Epochs 11](#_Toc205548337)

[Task 9: Implement Callbacks 12](#_Toc205548338)

[Task 10: Visualise the Training Process: 13](#_Toc205548339)

[Task 11: Tune Hyperparameters 14](#_Toc205548340)

[Task 12: Evaluate the Model on the Test Dataset: 14](#_Toc205548341)

|  |  |
| --- | --- |
| Assignment 3.1  ANN for Medical Diagnosis |  |

# Scenario and Instructions

#### **Scenario:**

A healthcare organisation wants to develop a diagnostic tool to predict whether a patient has diabetes based on various health metrics. Your task is to implement an Artificial Neural Network (ANN) to create an accurate prediction model that can assist healthcare providers in early diagnosis.

#### **Assignment Instructions:**

Create a Python notebook to construct, train, and evaluate an artificial neural network for diabetes prediction. This assignment will integrate supervised learning concepts with practical healthcare applications while exploring the capabilities of neural networks.

**Dataset**

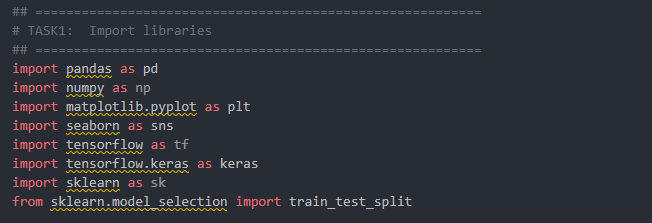
The [enhanced\_diabetes\_dataset.csv](https://classroom.emeritus.org/courses/12437/files/4675205?wrap=1) contains health information for female patients of Pima Indian heritage, including:

* Pregnancies: Number of times pregnant
* Glucose: Plasma glucose concentration (mg/dL)
* BloodPressure: Diastolic blood pressure (mm Hg)
* SkinThickness: Triceps skin fold thickness (mm)
* Insulin: 2-Hour serum insulin (mu U/ml)
* BMI: Body mass index (weight in kg/(height in m)²)
* DiabetesPedigreeFunction: Diabetes pedigree function (a function which scores likelihood of diabetes based on family history)
* Age: Age in years
* Outcome: Class variable (0 or 1) indicating whether the patient has diabetes (1) or not (0)

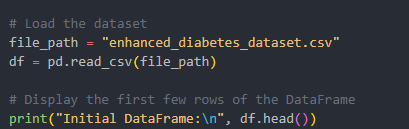
# Assignment & Solutions

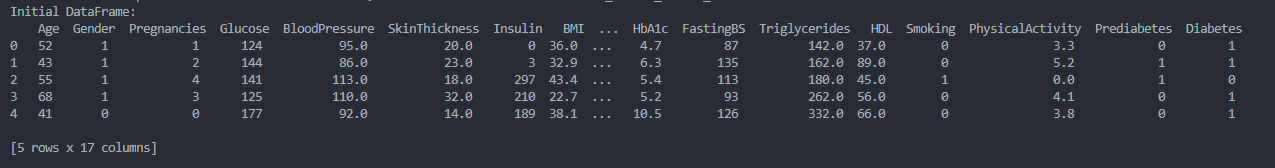
## Task 1: Utilise Libraries/Dataset

Import all necessary libraries (pandas, numpy, matplotlib, seaborn, tensorflow/keras, scikit-learn)

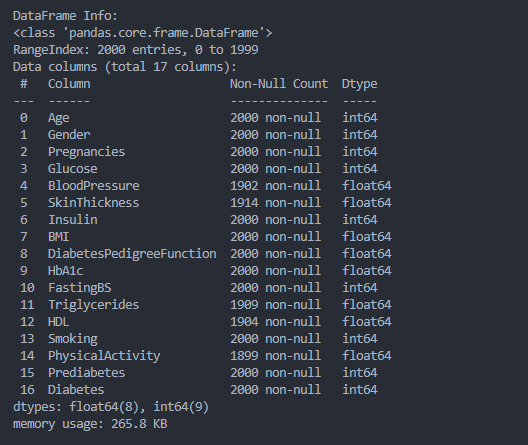


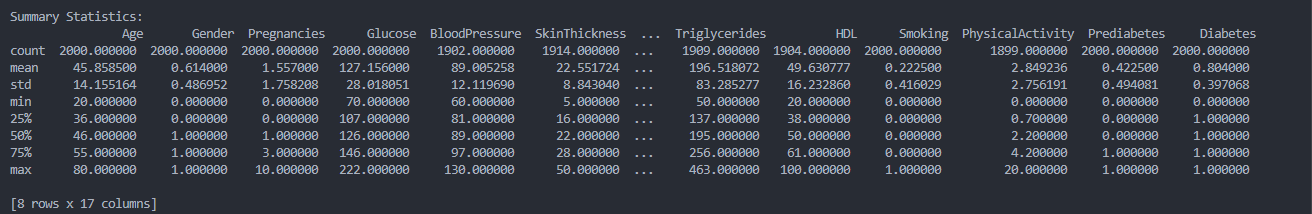
Load the diabetes dataset and display the first few rows





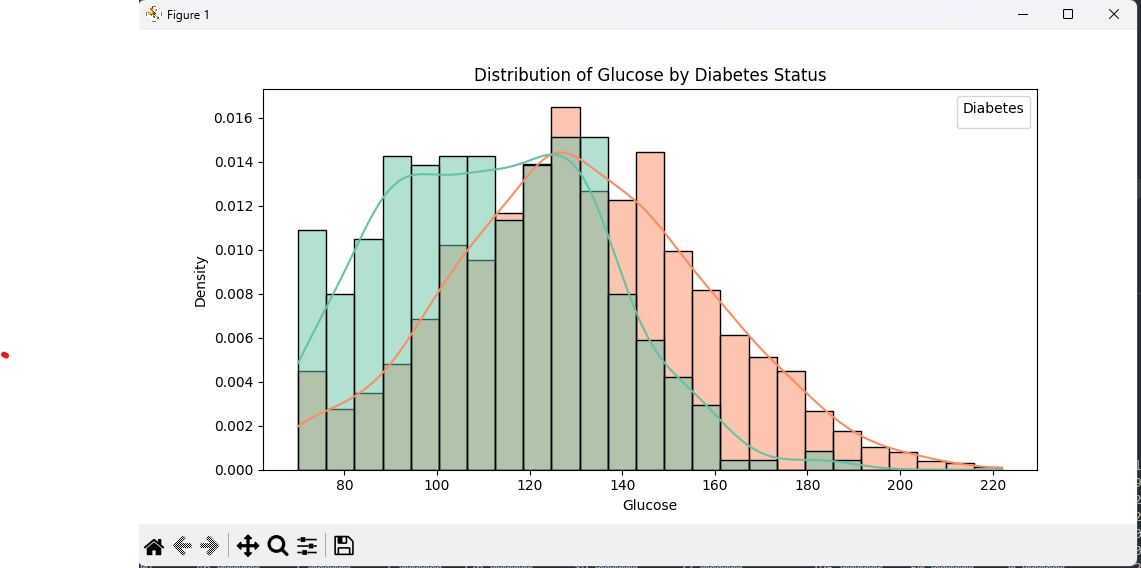
Show basic information and statistics about the dataset

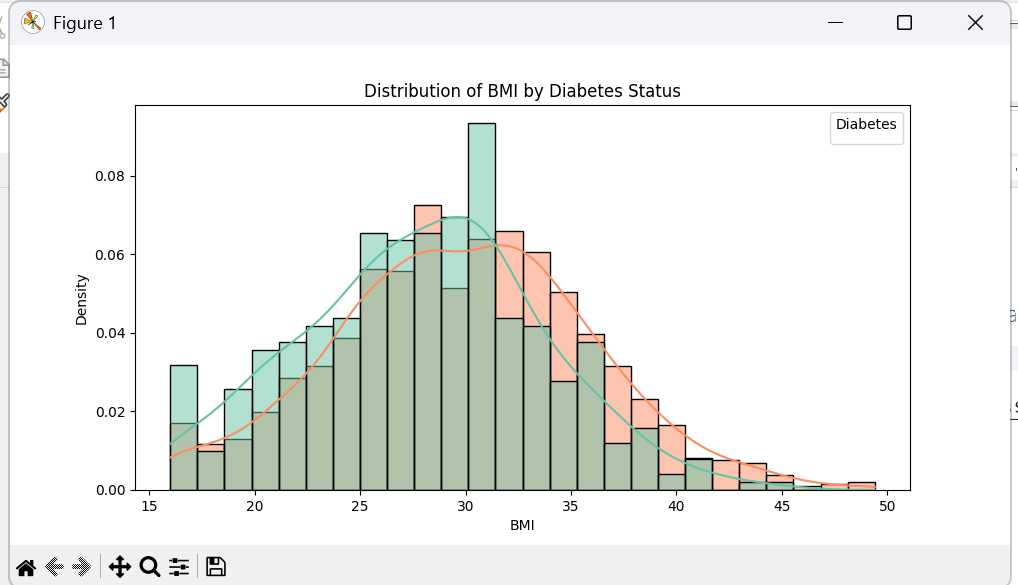


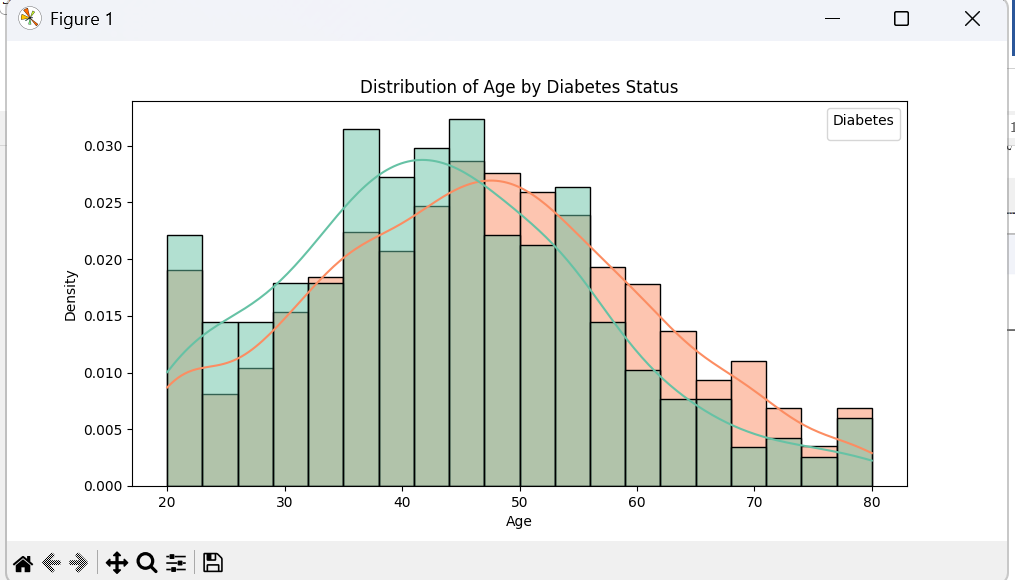


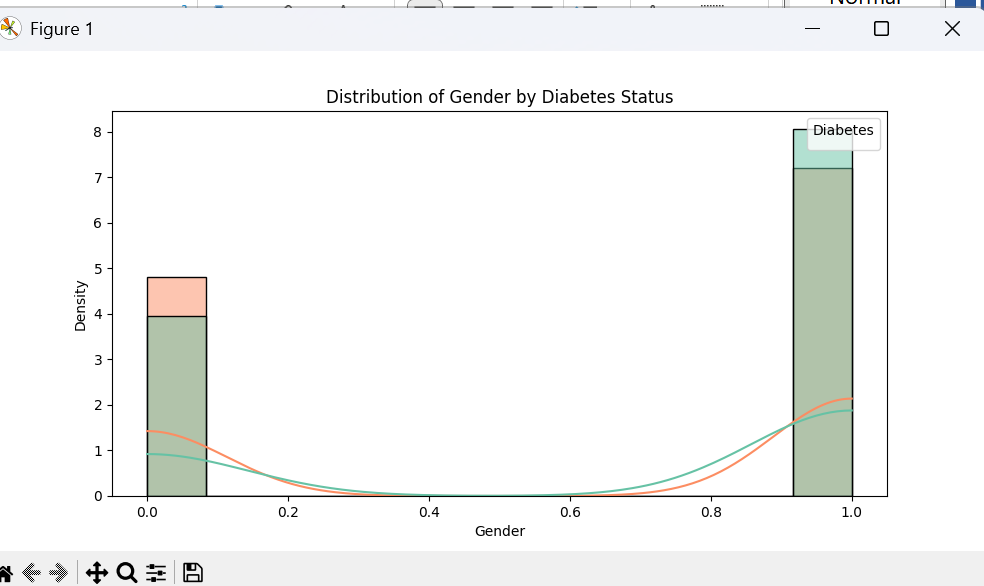
## Task 2: Generate at least three EDA visualisations

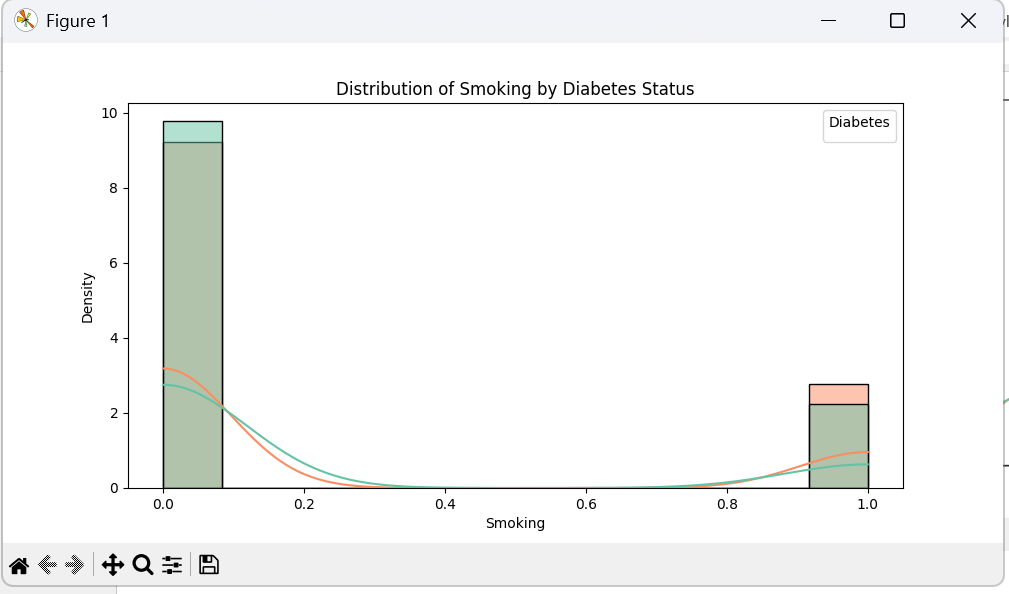
Create distribution plots comparing features between diabetic and non-diabetic patients

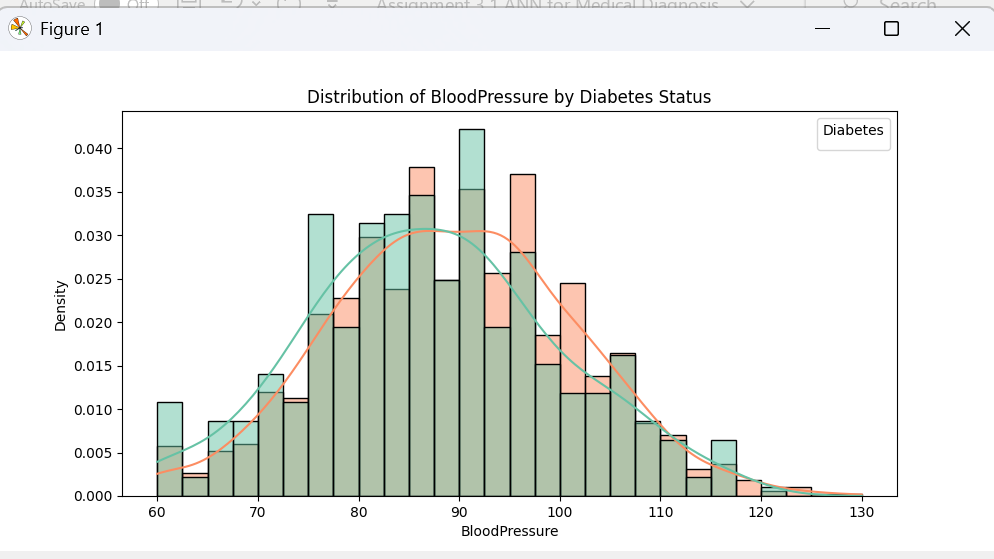


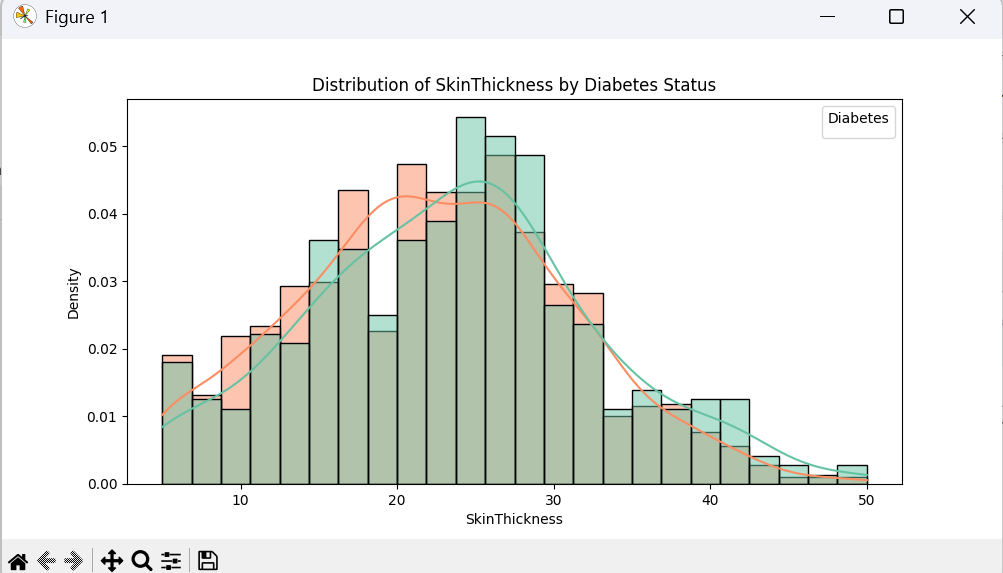




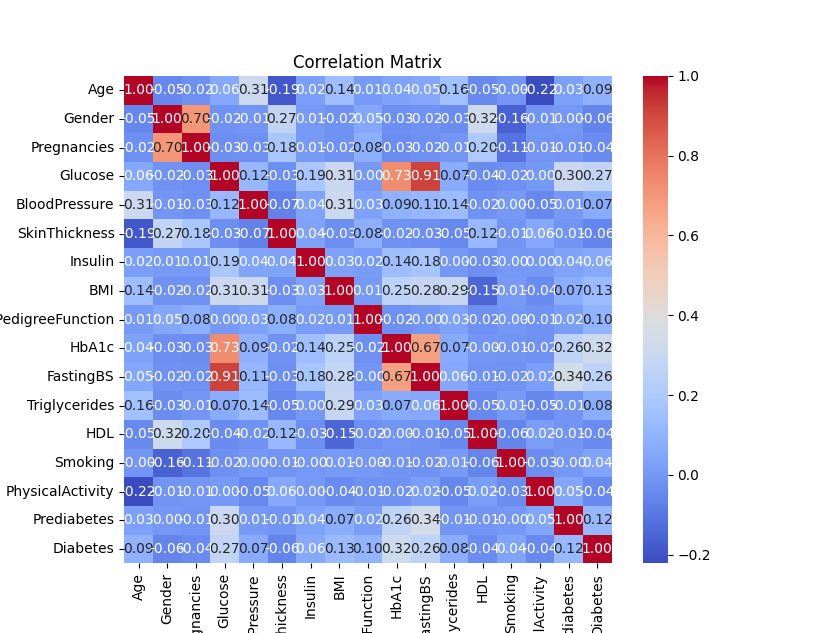




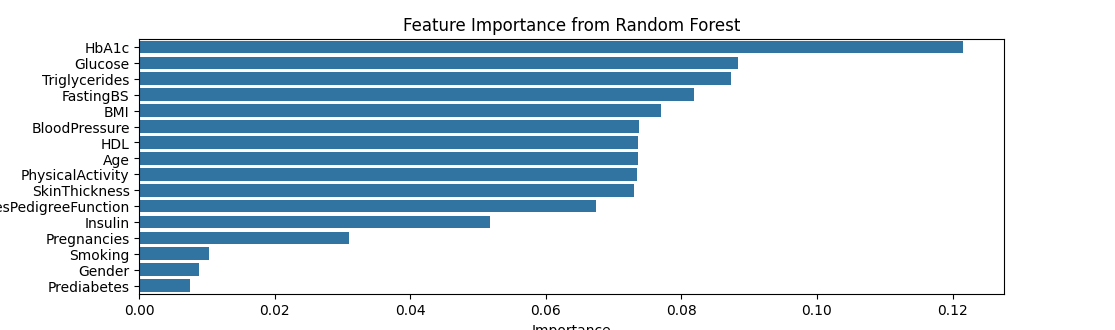




Develop correlation heatmap to identify relationships between features



Visualise feature importance using appropriate techniques ( Random Forest )



## Task 3: Analyse data quality

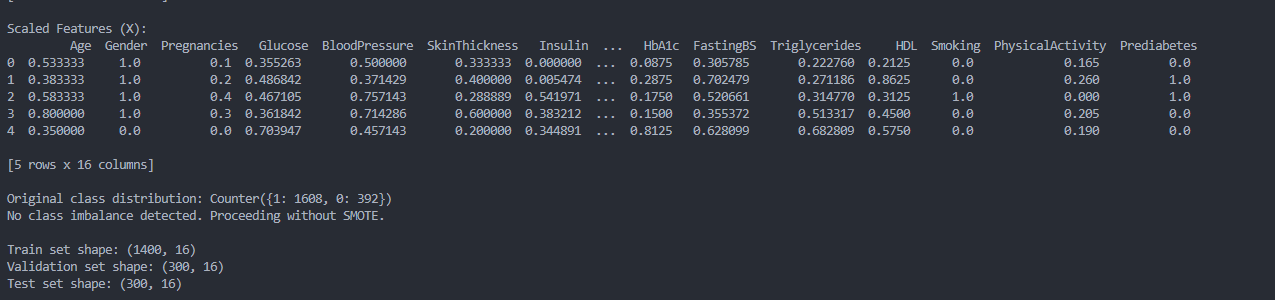
Check for missing or zero values (particularly in SkinThickness, Insulin, and BMI)

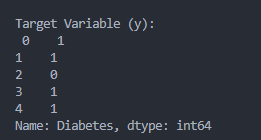
Output redirected to doc, attached herewith



## Task 4: Prepare Features and Target Variable

* Split data into features (X) and target (y)
* Normalise/standardise features

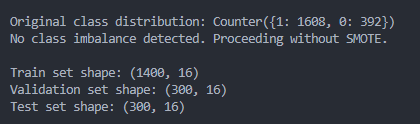




## **Task 5: Class Imbalance and Dataset Splitting**

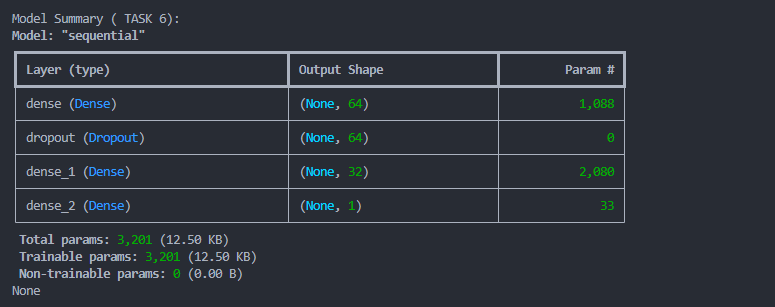
Handle class imbalance if present (using techniques like SMOTE or class weights).

Split data into training (70%), validation (15%), and test (15%) sets



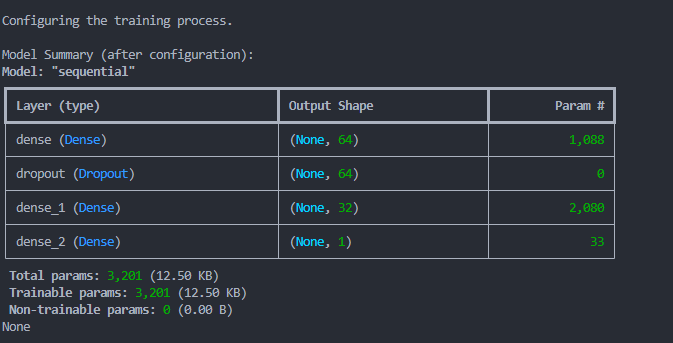
## Task 6: Design an Appropriate DCNN-based Architecture

* Determine appropriate number of layers and neurons
* Select suitable activation functions for hidden and output layers
* OPTIONAL - Justify your architectural choices
* DCNN ( Deep Convolutional Neural Network) is not suitable for data in table form. It is only suitable for images or videos
* Defined a simple neural network and printing the model summary

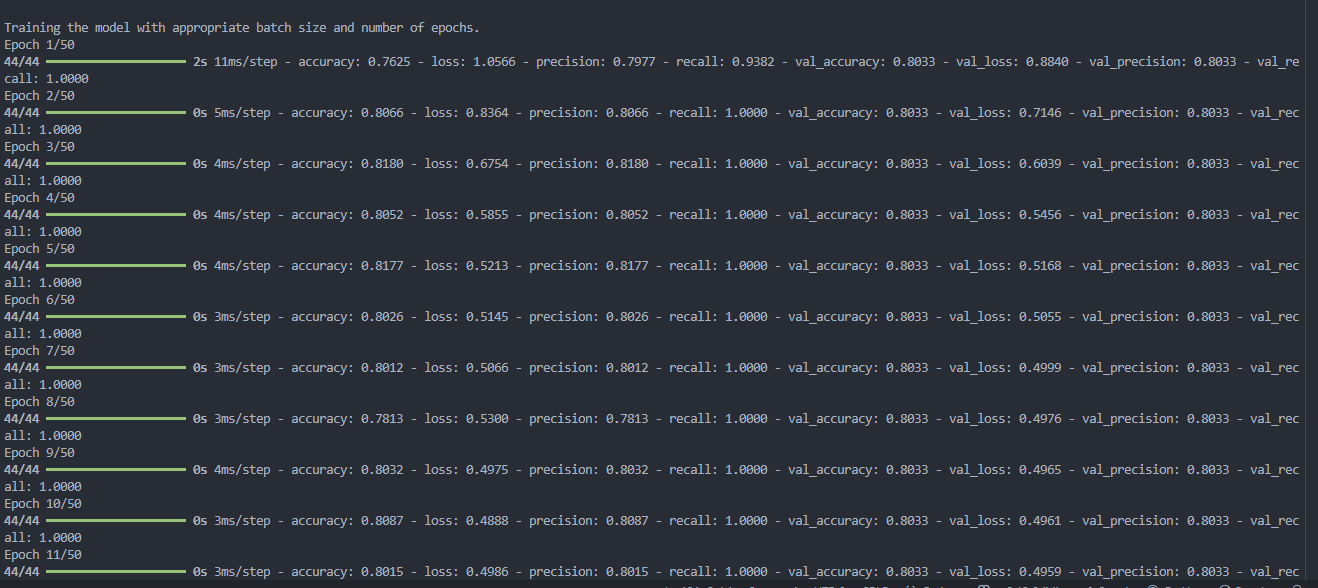


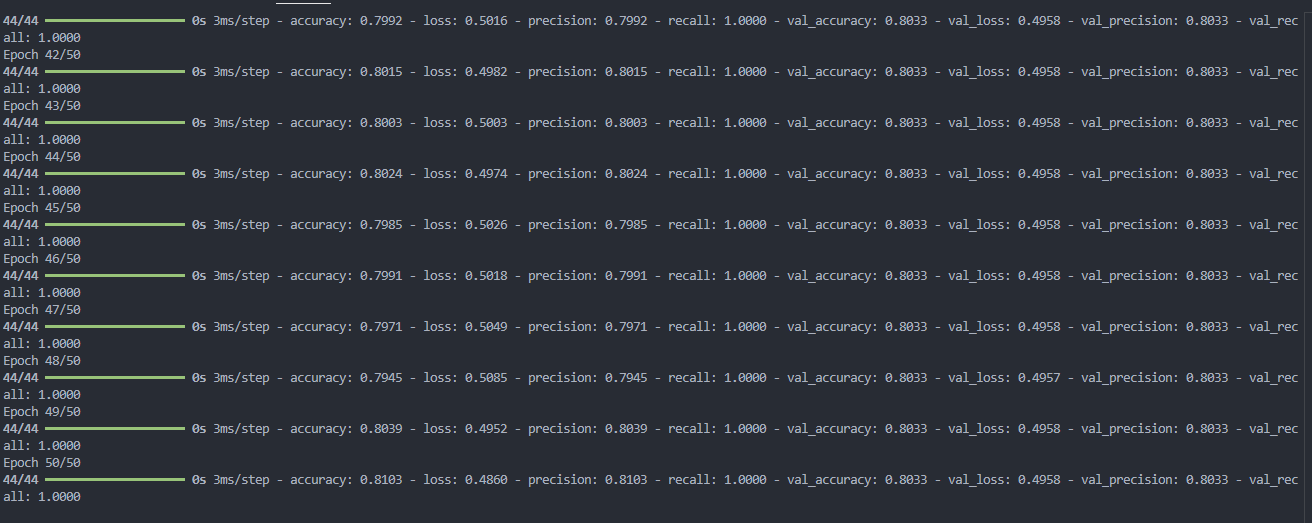
## Task 7: Configure the Training Process

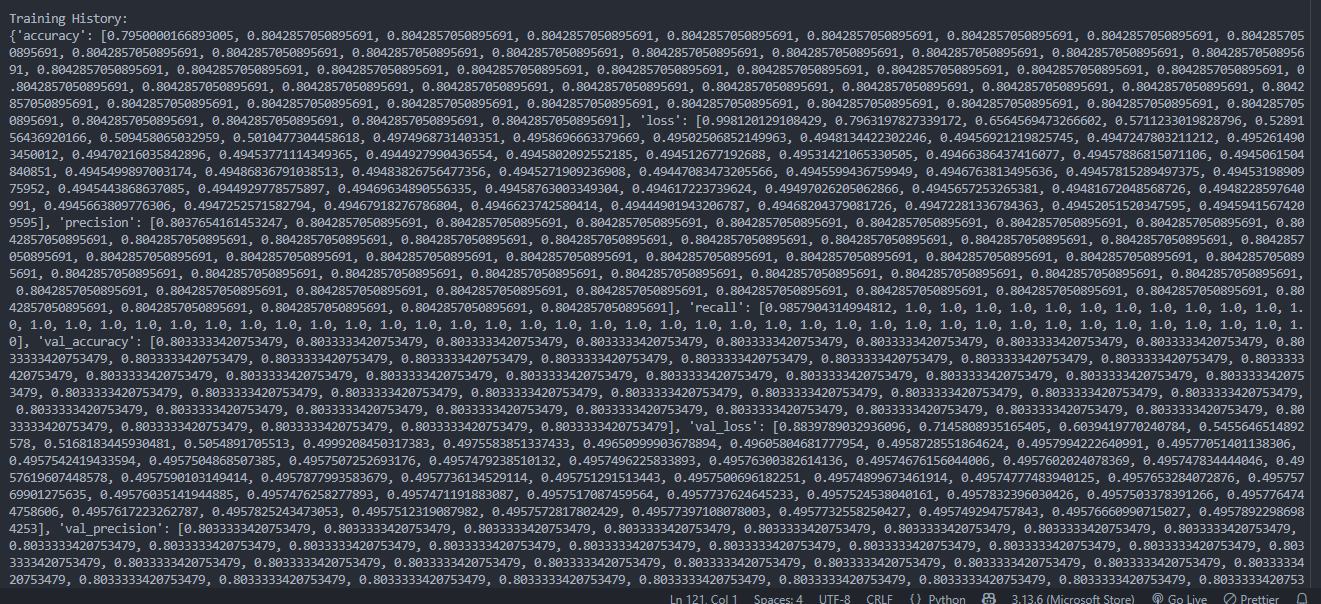
* Select appropriate loss function and evaluation metrics
* Choose optimiser and learning rate
* Implement regularisation techniques (dropout, L1/L2) to prevent overfitting
* Initialise and compile the model



## Task 8: Train the model with Appropriate Batch Size and Number of Epochs

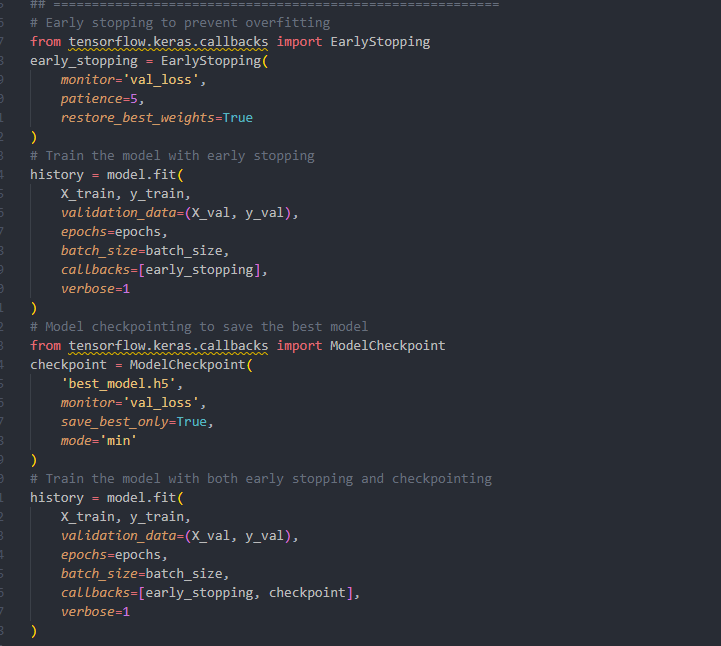






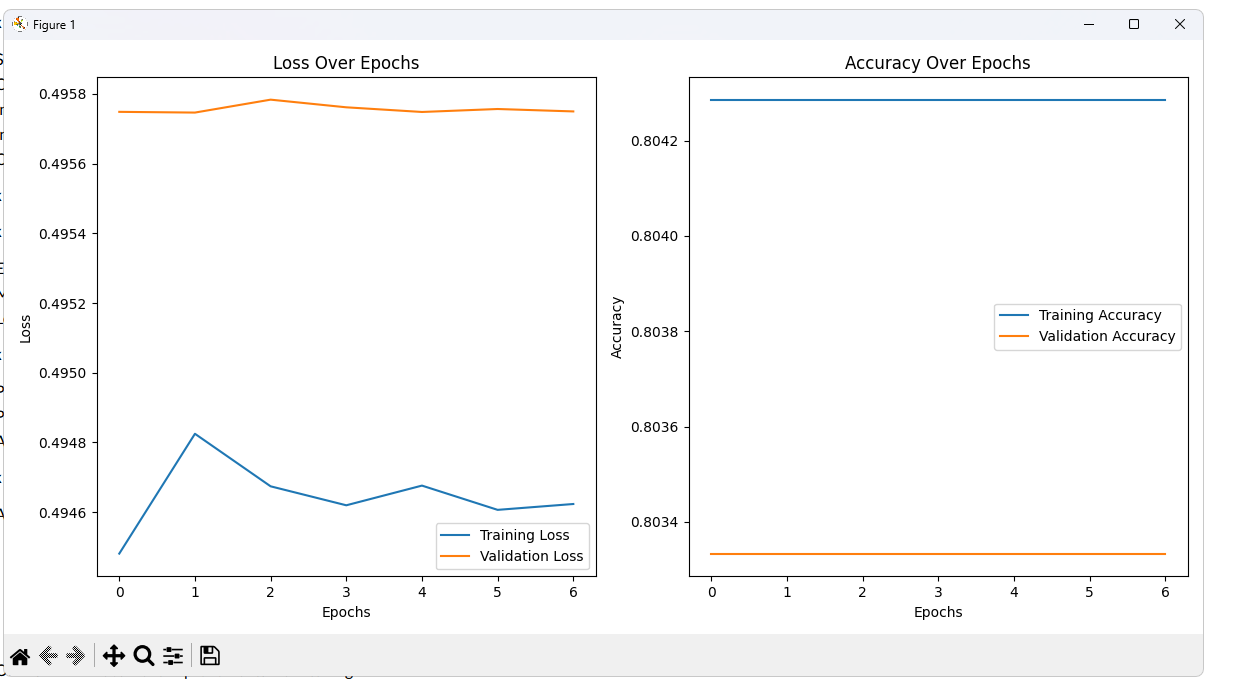
## Task 9: Implement Callbacks

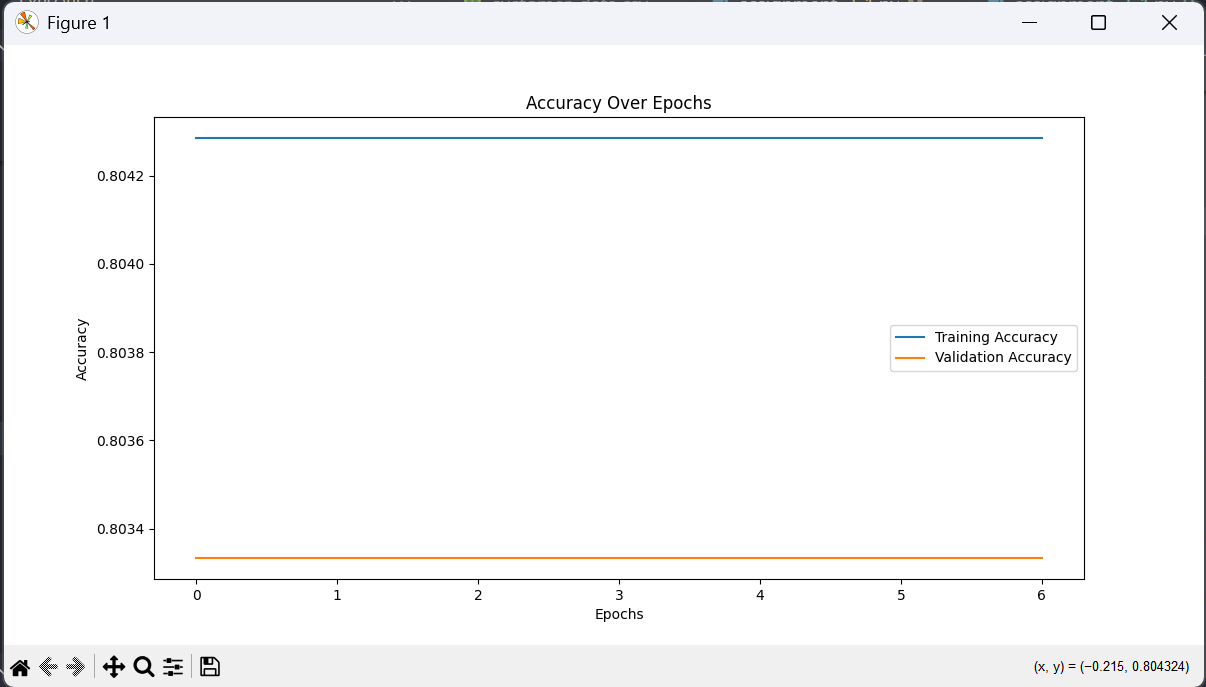
* Early stopping to prevent overfitting
* Model checkpointing to save the best model
* Learning rate scheduling if appropriate

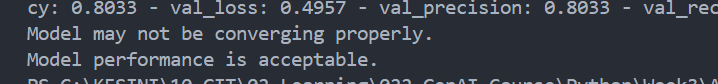


## Task 10: Visualise the Training Process:

* Plot training and validation loss
* Plot training and validation accuracy
* Analyse convergence and potential overfitting/underfitting







## Task 11: Tune Hyperparameters

* Adjust learning rate, batch size, or network architecture as needed. Use these values:
  + learning\_rates = [0.01, 0.001]
  + batch\_sizes = [16, 32]
  + dropout\_rates = [0.2]
  + filter\_numbers = [32, 64]
  + kernel\_sizes = [3]



## Task 12: Evaluate the Model on the Test Dataset:

* Calculate accuracy, precision, recall, F1-score
* Generate confusion matrix
* Create and analyse ROC curve and calculate AUC

