Deep Learning Assignment-1

Report on
Binary Classification Using Deep Neural Network For Predicting Heart Attacks

Group

1. **Rahul Bharat Harpal:** 22UCS158
2. **Dudhat Hemil Pravinbhai:** 22UCS074
3. **Siddharth Rajesh Jiyani:** 22UCS211

Submitted To
Dr. Preety Singh

Contents

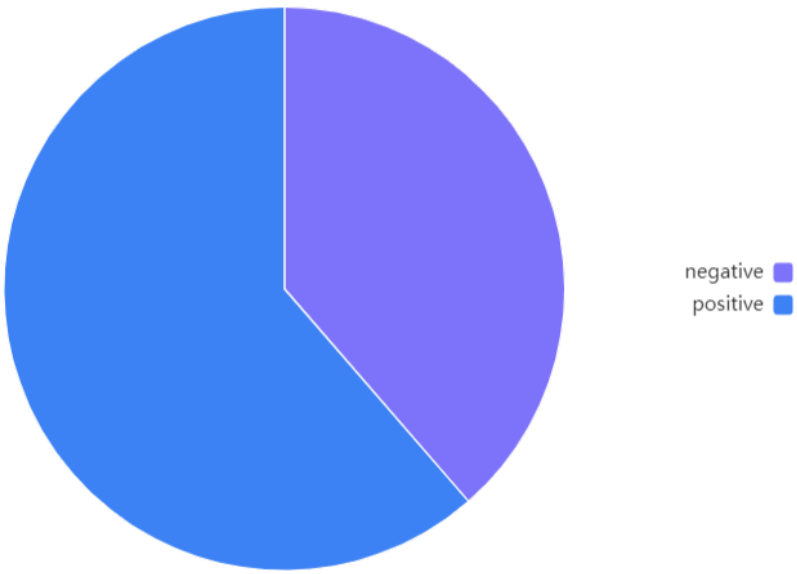
1. Introduction

In this assignment, we are going to train a Deep Neural Network model for predicting heart attacks using the Dataset taken from the website **Mendeley Data.**

- **Name: Heart Attack Dataset**
- **Link: https://data.mendeley.com/datasets/wmhctcrt5v/1 (We have removed some rows which are factually incorrect from this dataset; that dataset is present in the zip folder.)**
- **Contributors:** Tarik A. Rashid, Bryar Hassan
- **Description:** The heart attack datasets were collected at Zheen Hospital in Erbil, Iraq, from January 2019 to May 2019. The attributes of this dataset are age, gender, heart rate, systolic blood pressure, diastolic blood pressure, blood sugar, ck-mb and troponin with negative or positive output. According to the provided information, the medical dataset classifies either a heart attack or none. The gender column in the data is normalized: the male is set to 1 and the female to 0. As for the output, positive is set to 1 and negative to 0**.**

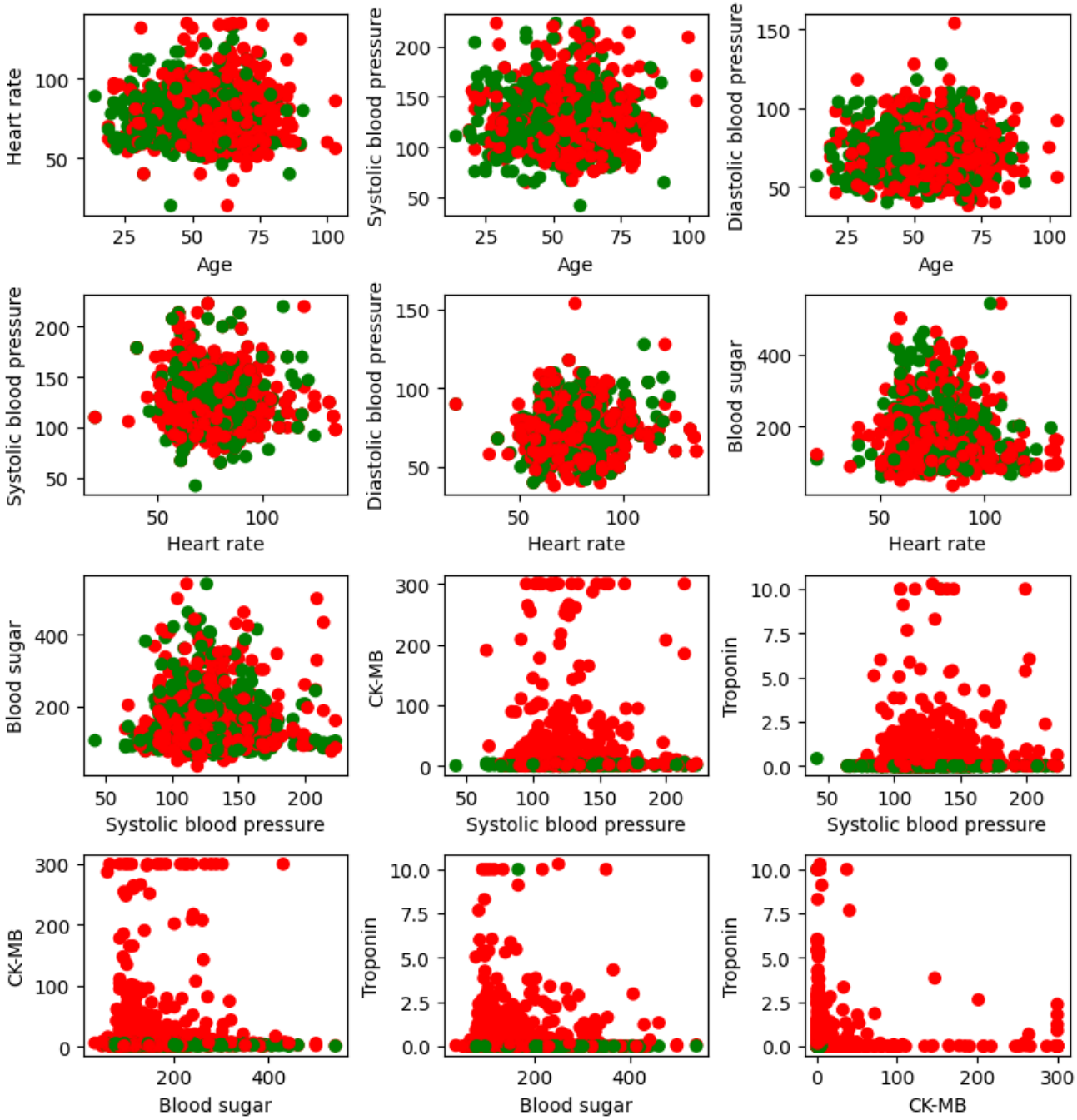
2. Visualizing the Dataset

- **Number of Samples:** 1319
- **Ground Truth: 61.4%** of the samples are positive for heart attack, and **38.6%** of samples are negative.

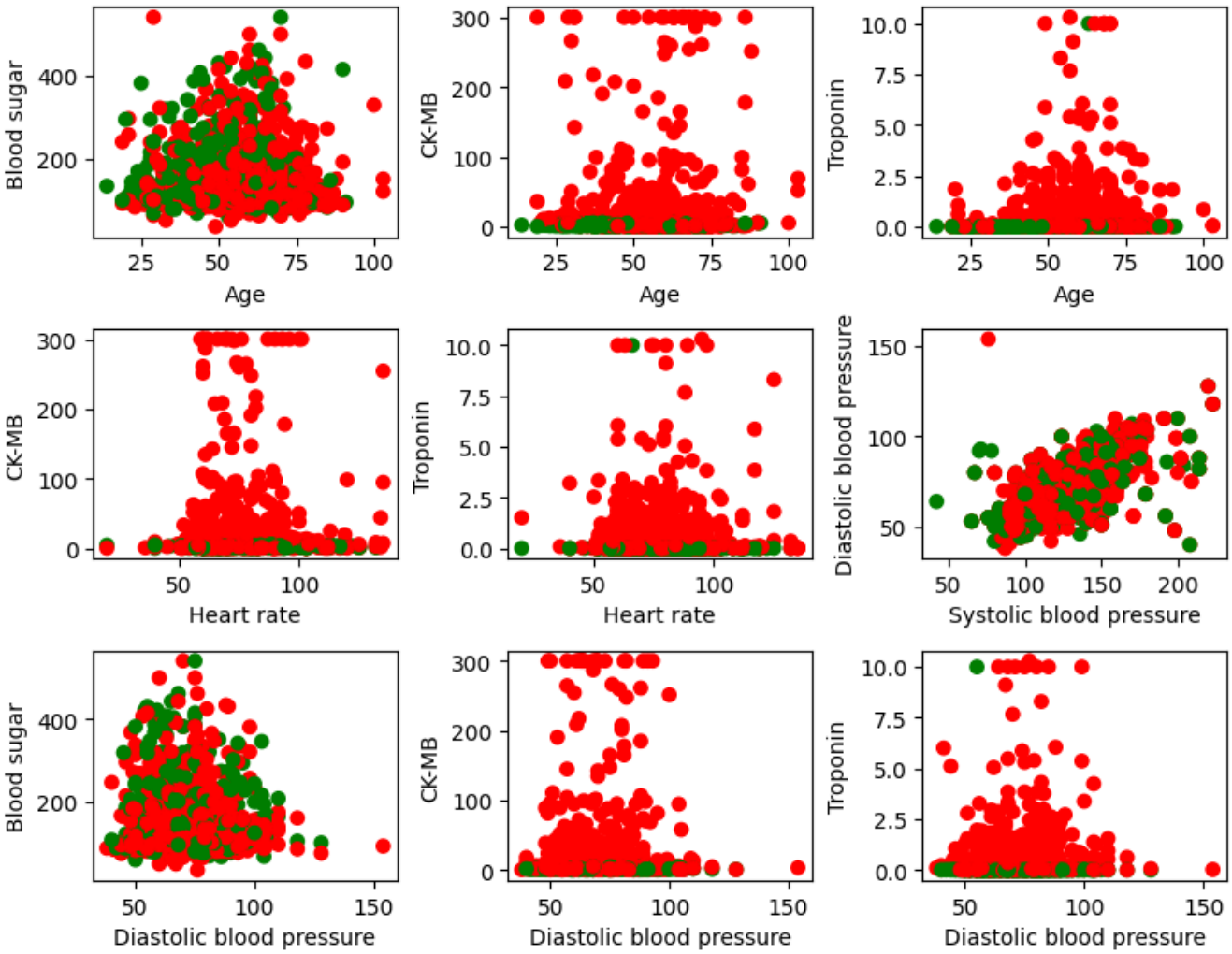


- **Here's a brief description of each feature:**
  1. **Age**: Between 14 to 103 years.
  2. **Gender**: 1 for male and 0 for female
  3. **Heart rate**: Measured in beats per minute(bpm). The number of times the heart beats per minute. It is a key measure of cardiovascular health, and abnormal values (either too high or too low) can be indicative of underlying conditions such as cardiovascular stress.
  4. **Systolic blood pressure**: Measured in millimetres of mercury (mmHg). The pressure in the arteries when the heart beats (contracts). High systolic pressure is a significant risk factor for heart disease and stroke and is often used to assess hypertension.
  5. **Diastolic blood pressure**: Measured in millimetres of mercury (mmHg). The pressure in the arteries between heartbeats (when the heart is resting). Elevated diastolic pressure is also linked to cardiovascular diseases and is important for assessing overall blood pressure health.
  6. **Blood sugar**: Measured in milligrams per deciliter (mg/dL). The concentration of glucose in the blood. Blood sugar levels are crucial for diagnosing diabetes and managing metabolic health, as both high and low levels can lead to serious complications.
  7. **CK-MB**: Measured in nanograms per millimetre (ng/mL). Creatine Kinase-MB is an enzyme found primarily in heart muscle cells. Elevated levels of CK-MB in the blood are often used to diagnose myocardial infarction (heart attack) and other cardiac injuries.
  8. **Troponin:** Measured in nanograms per millimetre (ng/mL). A protein involved in muscle contraction, specifically in the heart. Increased levels of troponin in the blood are a key indicator of heart muscle damage, often used to diagnose heart attacks.

- **Plotting Scatterplots:** Scatterplots are useful for identifying relationships or correlations between variables, such as linear trends or clusters. Patterns like positive, negative, or no correlation can be easily observed, making them a valuable tool for exploring data. We will plot these between each pair of features, and the dots on the plot are coloured based on whether the person suffered from a heart attack or not. Red dots mean positive for heart attack.

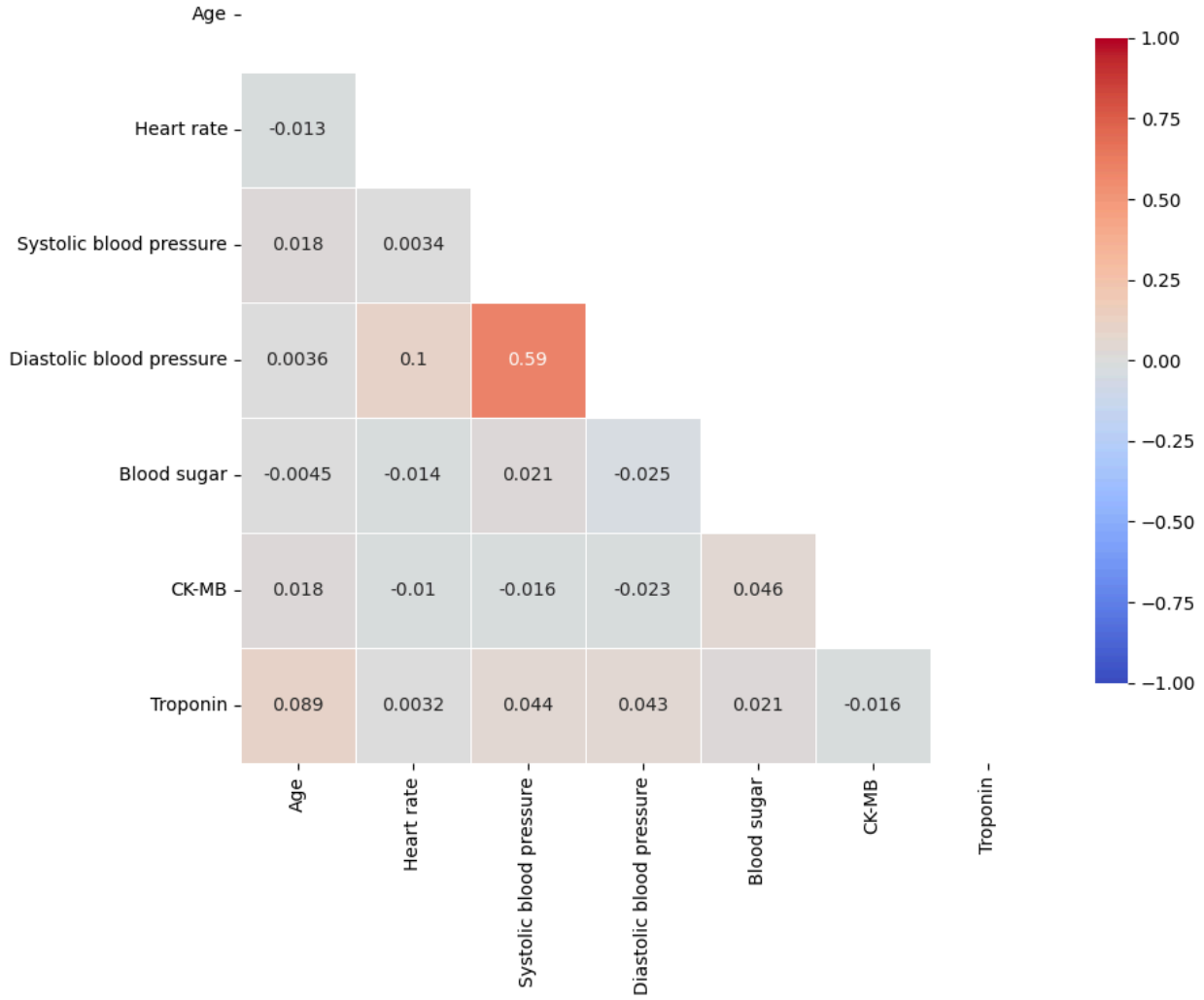


Scatterplots (Part 1)



Scatterplots (Part 2)

- **Computing the Correlation Matrix:** A correlation matrix is a table that displays the correlation coefficients between pairs of variables in a dataset. Each cell in the matrix shows the strength and direction of the relationship between two variables, with values ranging from -1 to 1. A value of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 means no correlation. This matrix helps quantify the relationship between features which may not be clearly visible via scatterplots.



Pair-wise correlation coefficient for all the continuous variables

3. Data Preprocessing

- **Data Loading**: The dataset is loaded using pandas from a CSV file named 'Medicaldataset.csv'.

- **Handling Missing Values**: Rows with missing values are removed using `data.dropna()`, to ensure that only complete records are used for training.

- **Label Encoding**: The target variable 'Result', which represents medical test outcomes as 'positive' or 'negative', is converted to numerical values (0 for 'negative' and 1 for 'positive')

- **Standardization**: Features such as 'Age', 'Gender', 'Heart rate', 'Systolic blood pressure', 'Diastolic blood pressure', 'Blood sugar', 'CK-MB', and 'Troponin' are standardized using StandardScaler to have a mean of 0 and standard deviation of 1.
- **Adding new features**:
    - **Polynomial Features**: Squared terms of 'CK-MB', 'Troponin', and 'Age' are added to the dataset as new features. To add non-linearity to our model and help the model in learning potential quadratic relationships
    - **Interaction Terms**: Interaction terms like 'CK_MB_Troponin' (product of 'CK-MB' and 'Troponin') are added because they both are markers of heart muscle damage, and 'Age_BloodSugar' (product of 'Age' and 'Blood sugar') because cardiovascular diseases occur more prominently with older age and high sugar levels.

Feature Matrix and Target Vector:

- `X` is the feature matrix (input features).
- `Y` is the target vector (medical test outcomes).

4. Model Architecture

- **Neural Network Structure**:
    - The model has the following layers:
        - **Input Layer**: The number of input neurons equals the number of features (derived from the preprocessing step).

    - **Hidden Layer 1**: 15 neurons, using the ReLU activation function.
    - **Hidden Layer 2**: 8 neurons, also using ReLU activation function.
    - **Output Layer**: 1 neuron, using the Sigmoid activation function to output probabilities (for binary classification).
- **Parameter Initialization**:
    - Weights are initialized using He initialization because it is most suitable for **ReLU** activation function (`np.sqrt(2./fan_in)`) to maintain variance through layers.
    - Biases are initialized to zeros.

Forward Propagation

- **ReLU Activation**: Used in the hidden layers to introduce non-linearity, defined as $ReLU(Z) = max(0, Z)$.
- **Sigmoid Activation**: Used in the output layer to produce a probability between 0 and 1, suitable for binary classification.
- **Dropout Regularization**: A dropout rate of 0.8 is applied during training to prevent overfitting, keeping neurons active with an 80% probability.

Cost Computation

- **Binary Cross-Entropy Loss**: The model's cost is computed using cross-entropy, appropriate for binary classification tasks:

  $$Loss = -1/m * \sum(Ylog(Y') + (1 - Y)log(1 - Y'))$$

  where Y' is the predicted probability, Y is the true label, and m is the number of samples in a mini-batch.

- **L2 Regularization**: A regularization term (λ) is added to the cost to prevent overfitting by penalizing large weights. m is the batch size.

  $$L2\ Regularization\ Cost = \lambda/2m * \sum W^2$$

Backward Propagation

- **Gradient Calculation**:
    - Gradients of the loss with respect to weights and biases are computed using derivatives of ReLU and Sigmoid functions.

$$\nabla_W L'(W) = \lambda * W + \nabla_W L(W)$$

    - Dropout adjustments ensure gradients only flow through active neurons.

- **Weight Updates**: Using gradient descent, the model updates its parameters:
  $W = W - learning\_rate \times dW$

  $b = b - learning\_rate \times db$

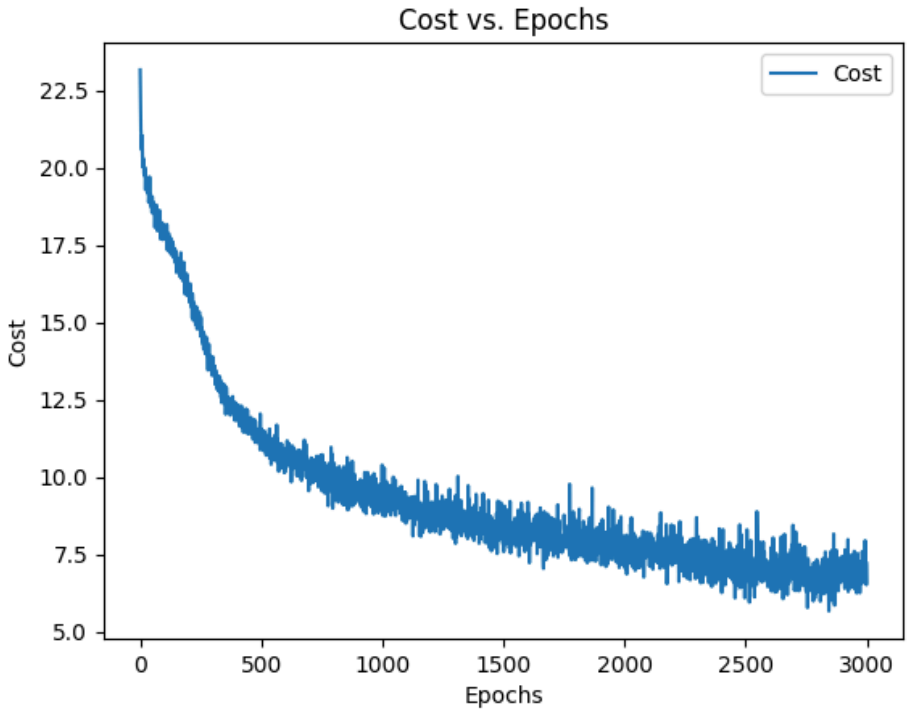
  where dW and db represent gradients of weights and biases, respectively.

5. Model Training

- **Training Data**: The data is split into training and testing sets by an 80:20 ratio. (1056 train samples and 264 test samples)
- **Batch Training**: Mini-batch gradient descent is implemented with a batch size of 32, where the training data is shuffled and processed in batches.
- **Epochs**: The model is trained over 3000 epochs, and the training cost is monitored every 100 epochs.
- **Evaluation Metrics**: Training and test accuracies are calculated periodically to monitor the model's performance.

6. Evaluation and Results

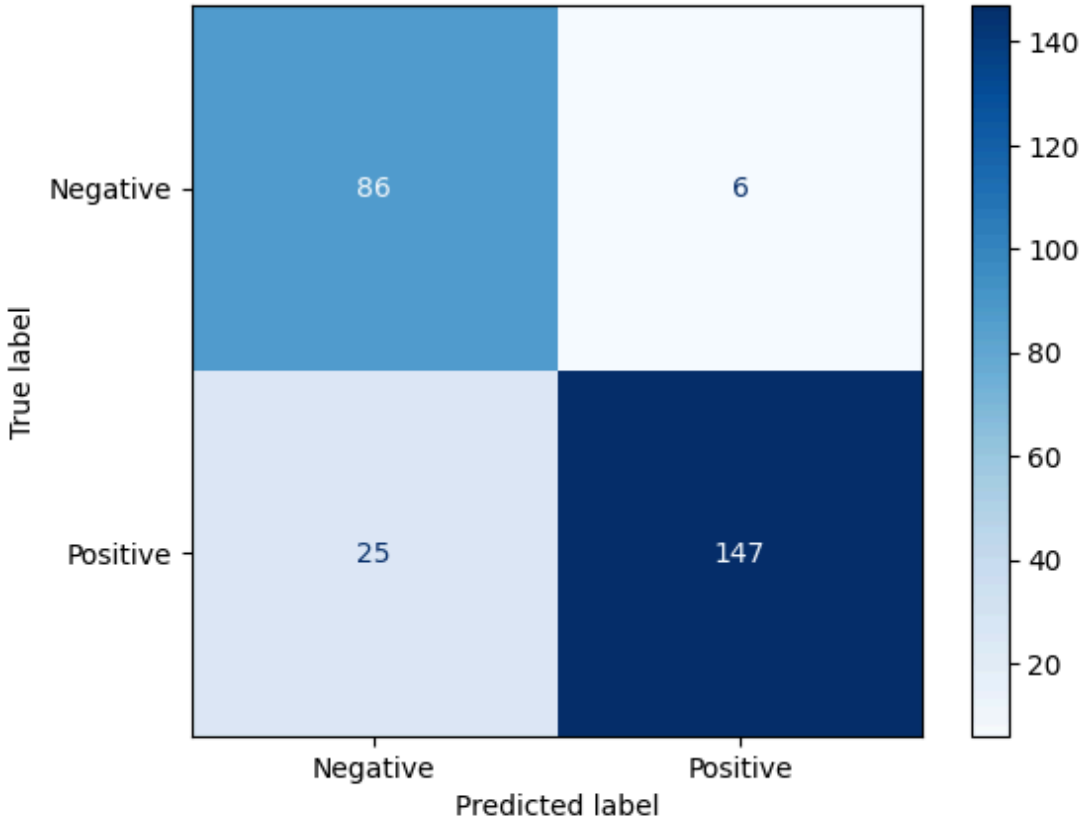
- **Cost v/s Epoch Plot**: Plot of costs plotted against every epoch



- **Final Accuracy**:
  - **Training Accuracy**: Calculated using predictions on the training set after the model is fully trained.
  - **Test Accuracy**: Evaluated using predictions on the test set, which measures the model's generalization capability.

```
Final Training Accuracy: 95.92%
Final Test Accuracy: 92.05%
```

- **Confusion Matrix**: Provides insight into true positive, true negative, false positive, and false negative counts, helping understand model performance in detail.



Confusion Matrix for 264 Test Samples

1. **True Positives:** 147/264
2. **True Negatives:** 86/264
3. **False Positives:** 6/264
4. **False Negatives:** 25/264

7. Conclusion

- The developed neural network model for medical test outcome prediction is satisfactorily accurate and stable. The model, with a learning rate of 0.01, under very intensive training for 3000 epochs, **has achieved a Final Training Accuracy of 95.92%**, which means it effectively learned the underlying pattern and relationships that may occur within a training data set. **A Final Test Accuracy of 92.05%** means the model has good generalizing properties when put into new data, making it suitable for real-world application in medical diagnostics. **Note: The accuracies and distribution on the confusion matrix may vary on every run because of some randomness due to dropout.**
- It involves relatively **high test accuracy, with dropout and L2 regularization techniques;** therefore, the model should not struggle with overfitting. More complex relations within data can be captured by including polynomial features and interaction terms. In general, this model can, therefore, turn out to be quite a good method for predicting patient outcomes from clinical data and may help medical experts make the right decisions.