Model Development Report

Model Objective: To build a model that correctly classifies if a student will dropout or not.

1.1 Data Splitting:
The target value was converted to a binary class where 1 represents if a students drops out and 0 to represent a student who did not drop out

The dataset was split into three subsets:

- Training Set: Used to train the models.

- Validation Set: Used to tune hyperparameters and assess model performance during training.

- Test Set: Held out for final evaluation to assess generalization ability.

The split was done using a standard 60%-20%-20% proportion for training, validation, and test sets, respectively. This ensures enough data is available for training while maintaining separate sets for hyperparameter tuning and unbiased performance evaluation. The train, validation and test set was scaled using the MinMaxScaling before training. A combination of the training and validation set was used as the dataset for cross validation to check how the model will perform in the real world. After the cross validation, the model was then evaluataed on the test set.

1.2 Baseline Model (Logistic Regression):

- Model Overview: Logistic regression was chosen as the baseline due to its simplicity and interpretability. It is a linear model that estimates the probability that an input belongs to a particular class.

- Training: The logistic regression model was trained on the training set using standard Scikit-learn implementation with default parameters.

- Hyperparameters: Regularization parameter (C), Penalty and solver were tuned using grid search and cross-validation.

- Evaluation: The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

Baseline Model Performance (Test Set):

- Accuracy: 87%

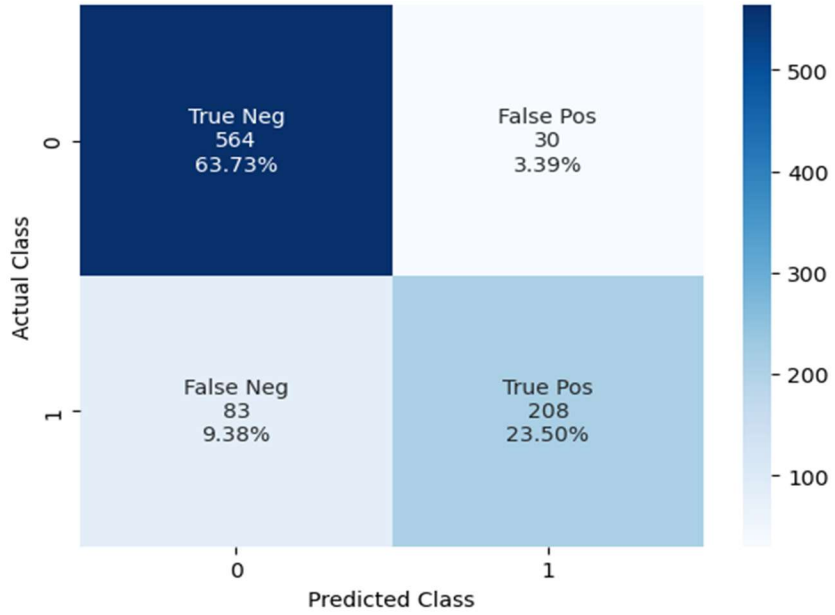
- Precision: 87%

- Recall: 83%

- F1-Score: 85

- ROC-AUC: 92%

Confusion Matrix:



The baseline performance provides a reference point to compare against more complex models.

2. Traditional Machine Learning Models

In this section, several traditional machine learning models were trained and evaluated:

2.1 Decision Trees

- Model Overview: A decision tree was used to model data by recursively splitting based on the feature that maximizes information gain (or minimizes Gini impurity).

- Hyperparameters: Maximum depth, maximum features, criterion and minimum samples per leaf were tuned using grid search and cross-validation.

- Evaluation: Despite being interpretable, decision trees are prone to overfitting. Regularization techniques like pruning were explored to control model complexity. The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

Performance (Test Set):

- Accuracy: 83%

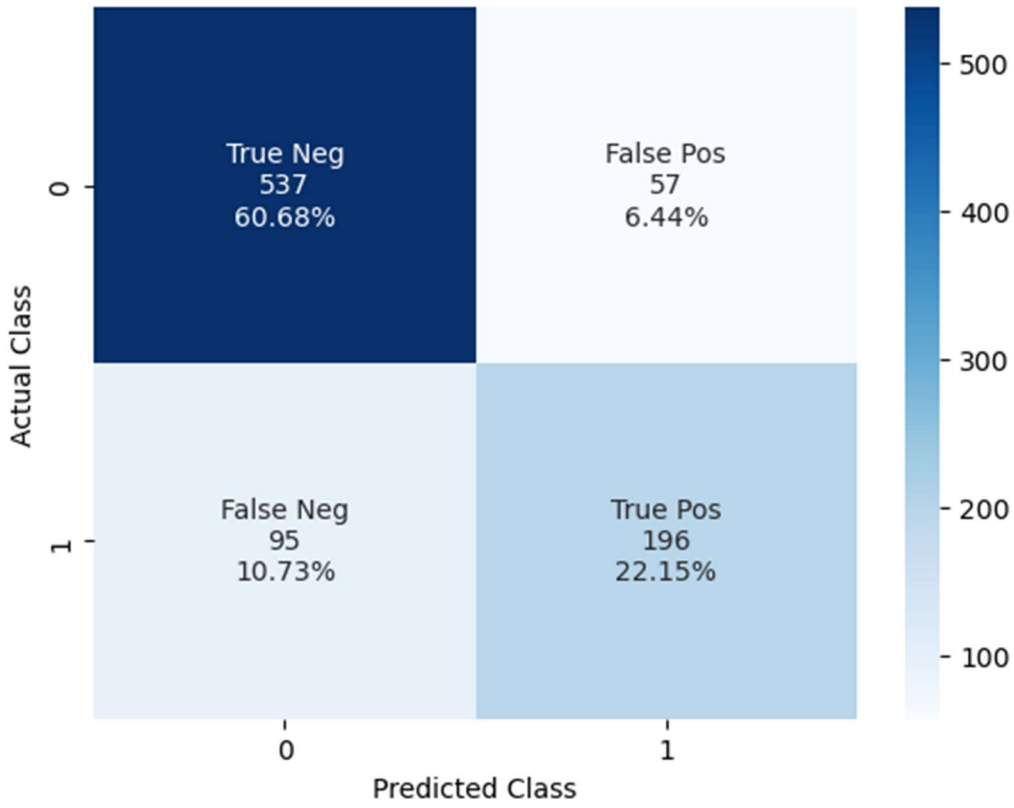
- Precision: 81%

- Recall: 79%

- F1-Score: 80

- ROC-AUC: 85%

Confusion Matrix:



2.2 Random Forests

- Model Overview: A random forest is an ensemble of decision trees. It reduces overfitting by averaging the predictions from multiple trees trained on different subsets of the data.

- Hyperparameters: Number of trees, minimum samples per leaf and maximum features were tuned.

- Evaluation: Random forests generally outperform single decision trees by reducing variance while retaining interpretability through feature importance metrics. The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

Performance (Test Set:

- Accuracy: 85%

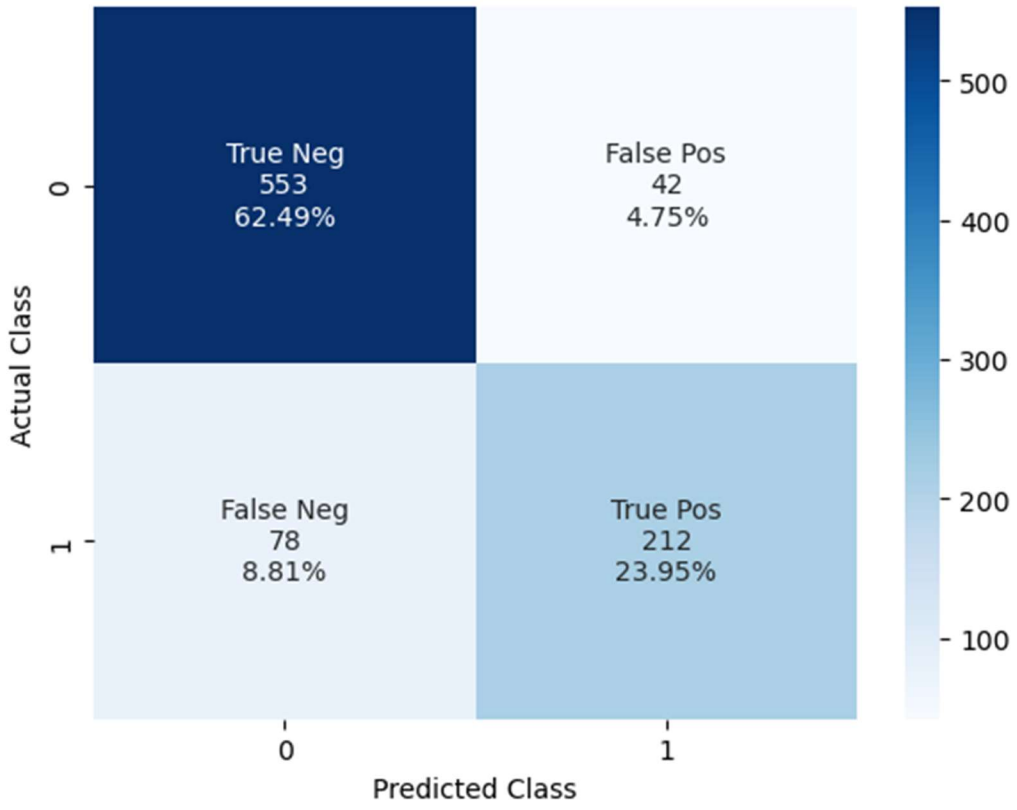
- Precision: 85%

- Recall: 80%

- F1-Score: 82

- ROC-AUC: 90%

Confusion Matrix:



2.3 Support Vector Machines (SVM)

- Model Overview: SVMs find the hyperplane that maximally separates the data points of different classes. For non-linear data, the kernel trick was used to project data into a higher-dimensional space.

- Hyperparameters: The choice of kernel (linear, RBF), gamma and regularization parameter (C) were tuned.

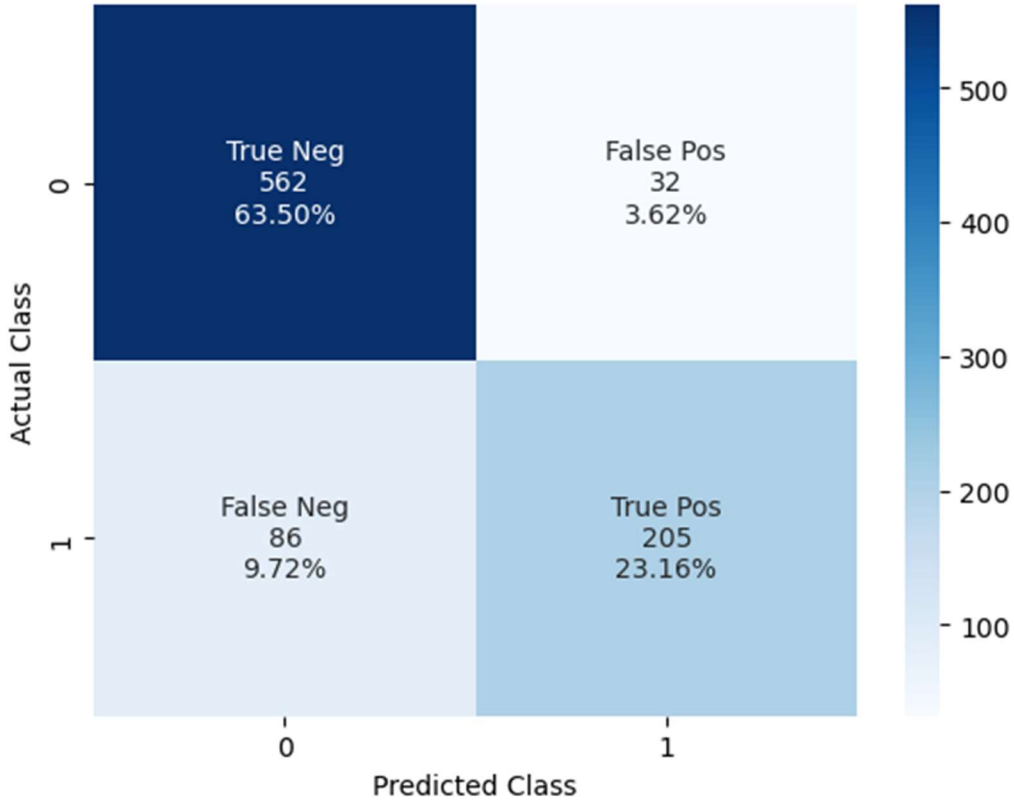
- Evaluation: SVMs were evaluated for their robustness to noisy data and ability to generalize well. However, they tend to be computationally expensive, especially

for large datasets. The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

Performance (Test Set):

- Accuracy: 87%
- Precision: 87%
- Recall: 83%
- F1-Score: 84
- ROC-AUC: 90%

Confusion Matrix:



2.4 Gradient Boosting Algorithm

- Model Overview: Gradient Boosting is a boosting algorithm that iteratively trains decision trees by focusing on errors made by previous trees. This allows for highly accurate predictions, especially on structured data.

- Hyperparameters: The number of trees, maximum depth, minimum samples split per leaf and subsamples were optimized.

- Evaluation: Gradient Boost typically provides state-of-the-art performance on many classification tasks but requires careful tuning to avoid overfitting. The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

Performance (Test Set):

- Accuracy: 86%

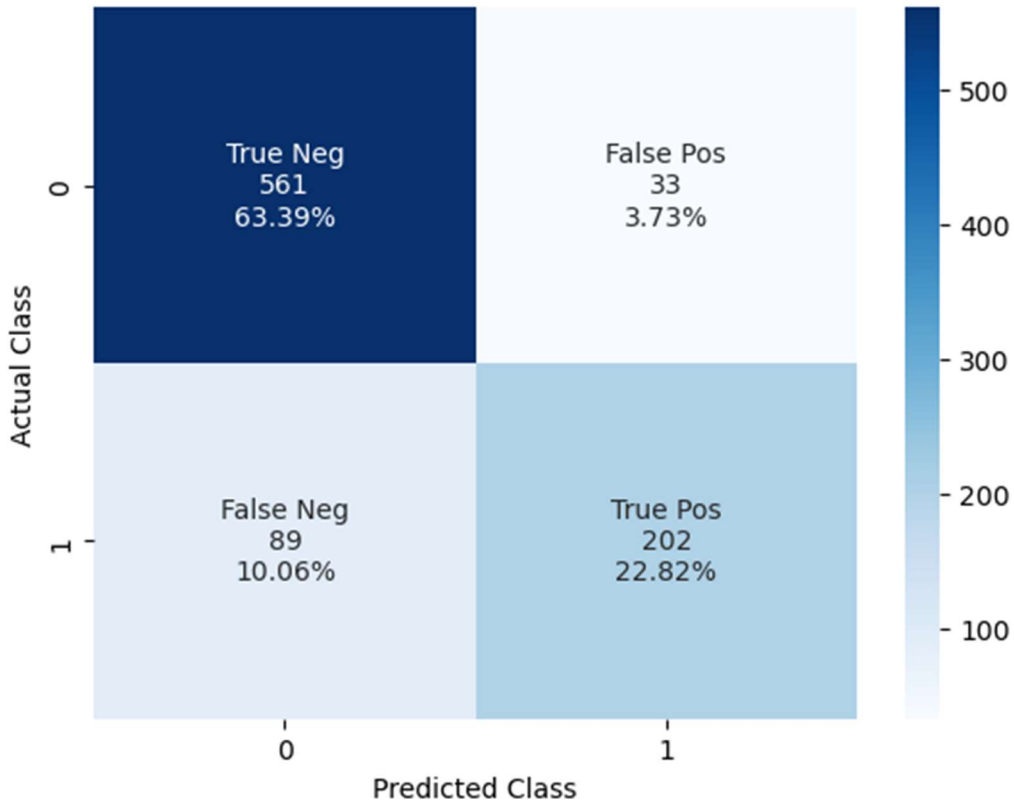
- Precision: 86%

- Recall: 82%

- F1-Score: 83

- ROC-AUC: 91%

Confusion Matrix:



2.5 Cat Boosting Machines

- Model Overview: Gradient Boosting is a boosting algorithm that iteratively trains decision trees sequentially by focusing on errors made by previous trees. Regularization techniques are integrated to control model complexity and enhance generalization.

- Hyperparameters: The number of iterations, depth, learning rate, border count etc were optimized.

- Evaluation: Cat Boost typically provides state-of-the-art performance on many classification tasks due to how it handles categorical data. The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

Performance (Test Set):

- Accuracy: 87%

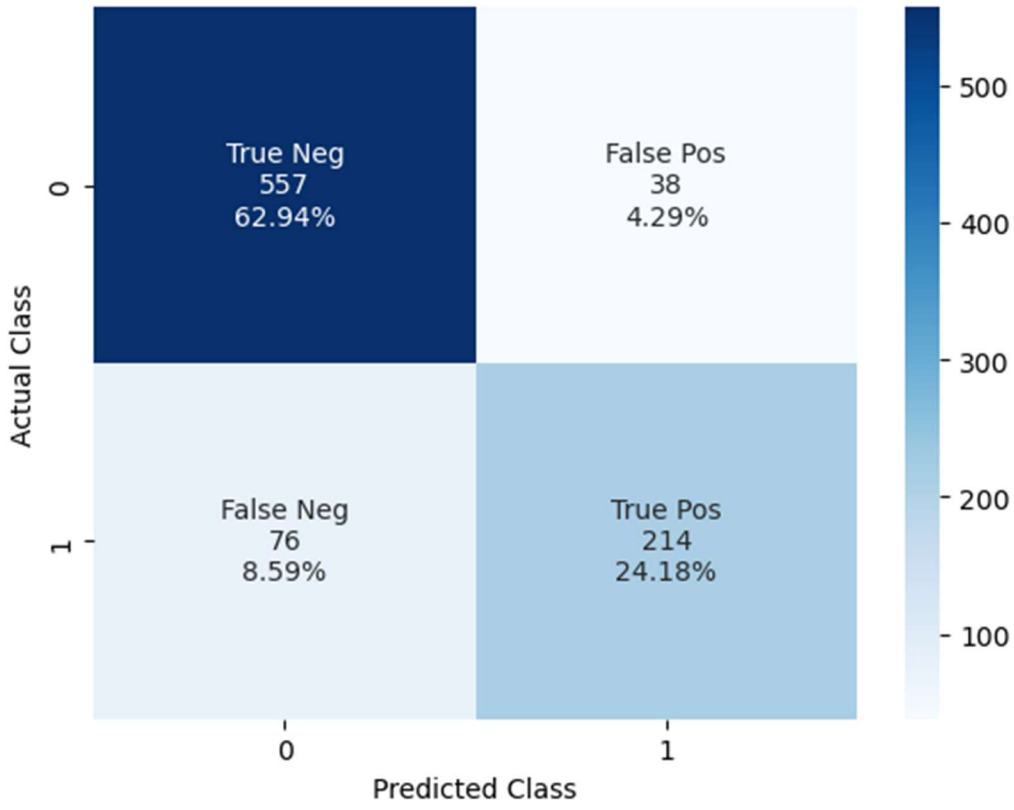
- Precision: 87%

- Recall: 82%

- F1-Score: 84

- ROC-AUC: 92%

Confusion Matrix:



2.5 Extreme Gradient Boosting

- Model Overview: Extreme Gradient Boosting is a boosting algorithm that iteratively trains decision trees sequentially by focusing on errors made by previous trees. It is an improvement to the gradient boosting algorithm.

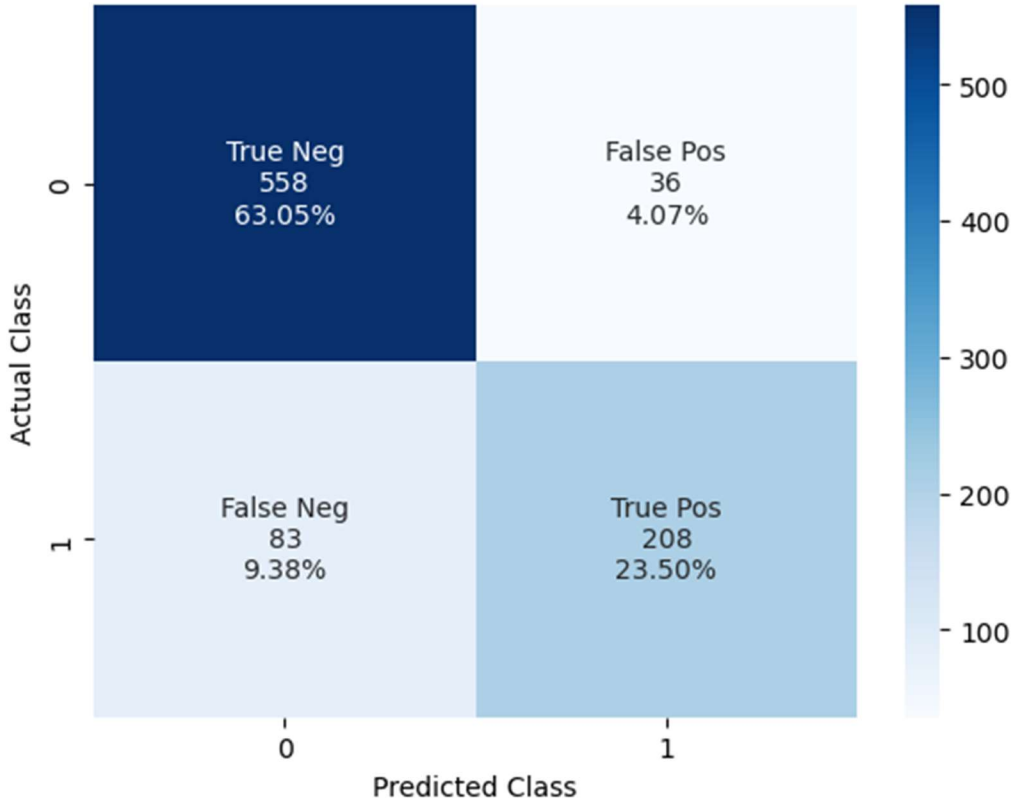
- Hyperparameters: The number of estimators, maximum depth, learning rate, subsample etc were optimized.

- Evaluation: The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

Performance (Test Set):

- Accuracy: 87%
- Precision: 86%
- Recall: 83%
- F1-Score: 84
- ROC-AUC: 92%

Confusion Matrix:



3. Deep Learning Models

3.1 Neural Network Architecture

- Model Overview: A fully connected neural network (feedforward) was designed to handle the classification task. The model consists of the following layers:

1. **Input Layer**:
    - **Layer Type**: Dense (Fully Connected)
    - **Number of Neurons**: 256
    - **Activation Function**: ReLU (Rectified Linear Unit)
    - **Input Shape**: The input layer accepts a feature vector with a shape equal to the number of independent variables.
    - **Description**: This layer serves as the entry point for the input data and applies the ReLU activation function to introduce non-linearity into the model. The 256 neurons allow the model to learn complex patterns from the input data.
2. **Hidden Layers**:
    - **Layer 2**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 128
        - **Activation Function**: ReLU
        - **Description**: The second layer further processes the features learned from the input layer. It retains the ReLU activation to maintain non-linearity and uses 128 neurons to capture more intricate relationships in the data.
    - **Layer 3**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 64
        - **Activation Function**: ReLU
        - **Description**: This layer continues to refine the feature representations, employing 64 neurons for deeper learning while utilizing the ReLU activation function to enhance performance.

- **Layer 4**:
  - **Layer Type**: Dense
  - **Number of Neurons**: 32
  - **Activation Function**: ReLU
  - **Description**: The fourth layer consists of 32 neurons, allowing the model to learn additional hierarchical features. The use of the ReLU activation function persists to provide non-linear transformations.
- **Layer 5**:
  - **Layer Type**: Dense
  - **Number of Neurons**: 16
  - **Activation Function**: ReLU
  - **Description**: This layer, with 16 neurons, further abstracts the learned features. It retains the ReLU activation to ensure effective learning of complex patterns.
- **Layer 6**:
  - **Layer Type**: Dense
  - **Number of Neurons**: 8
  - **Activation Function**: ReLU
  - **Description**: The sixth layer uses 8 neurons to compress the feature representation, allowing the model to focus on the most salient features. The ReLU activation function continues to facilitate non-linearity.

3. **Output Layer**:
   - **Layer Type**: Dense
   - **Number of Neurons**: 1
   - **Activation Function**: Sigmoid

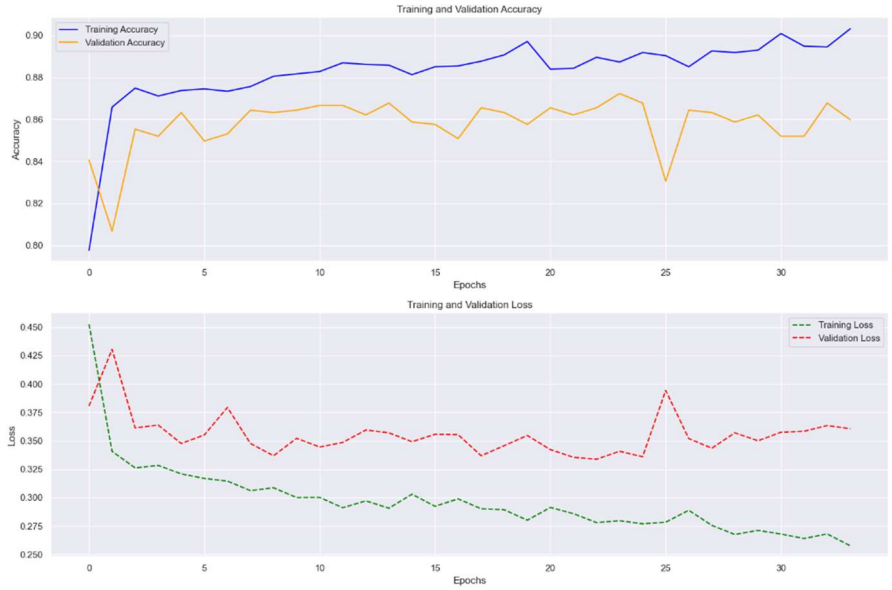
- **Description**: The final layer has a single neuron with a sigmoid activation function. This structure is standard for binary classification problems, as it outputs a probability score between 0 and 1, indicating the likelihood of the positive class.

3.2 Training the Neural Network

- Training: The network was trained using the Adam optimizer and binary cross-entropy loss function for binary classification.

- Evaluation: The model was evaluated on the validation and test sets. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were computed to assess its performance.

- Learning curve analysis:

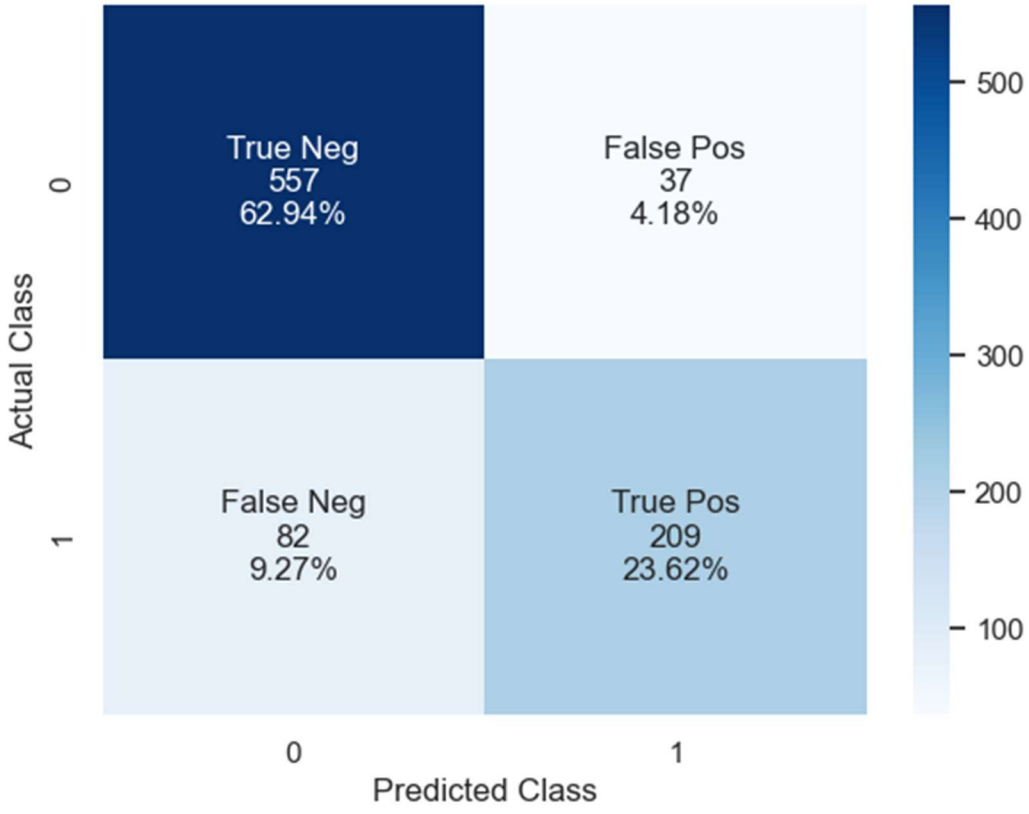


The learning curves indicate the model is well-trained with low bias (training accuracy ~0.88 and steadily decreasing training loss). However, it exhibits moderate variance, as seen in the fluctuating validation accuracy (~0.86) and higher, inconsistent validation loss compared to the training loss. This suggests slight overfitting.

Performance (Test Set):

- Accuracy: 87%
- Precision: 86%
- Recall: 83%
- F1-Score: 84
- ROC-AUC: 91%

Confusion Matrix:



Experimentation

Different architectures were tested by varying:

- Number of layers.
- Number of neurons per layer.

Training: All experimented deep learning models were trained using the Adam optimizer and binary cross-entropy loss function for binary classification.

Evaluation: These model was evaluated on accuracy and ROC-AUC. The training and validation loss were monitored to ensure there was no overfitting.

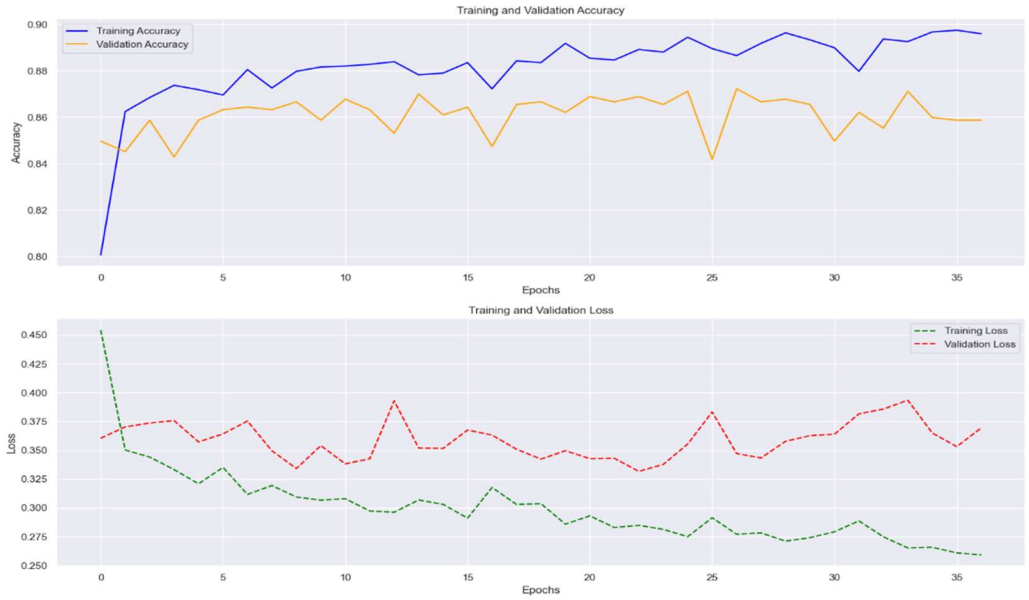
Model 1

The model consists of the following layers:

1. **Input Layer**:
    - **Layer Type**: Dense (Fully Connected)
    - **Number of Neurons**: 2048
    - **Activation Function**: ReLU (Rectified Linear Unit)
    - **Input Shape**: The input layer accepts a feature vector with a shape equal to the number of independent variables.
2. **Hidden Layers**:
    - **Layer 2**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 1024
        - **Activation Function**: ReLU
    - **Layer 3**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 512
        - **Activation Function**: ReLU
    - **Layer 4**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 128
        - **Activation Function**: ReLU
    - **Layer 5**:

        - **Layer Type**: Dense
        - **Number of Neurons**: 64
        - **Activation Function**: ReLU
    - **Layer 6**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 8
        - **Activation Function**: ReLU
3. **Output Layer**:
    - **Layer Type**: Dense
    - **Number of Neurons**: 1
    - **Activation Function**: Sigmoid

Learning Curve Analysis:

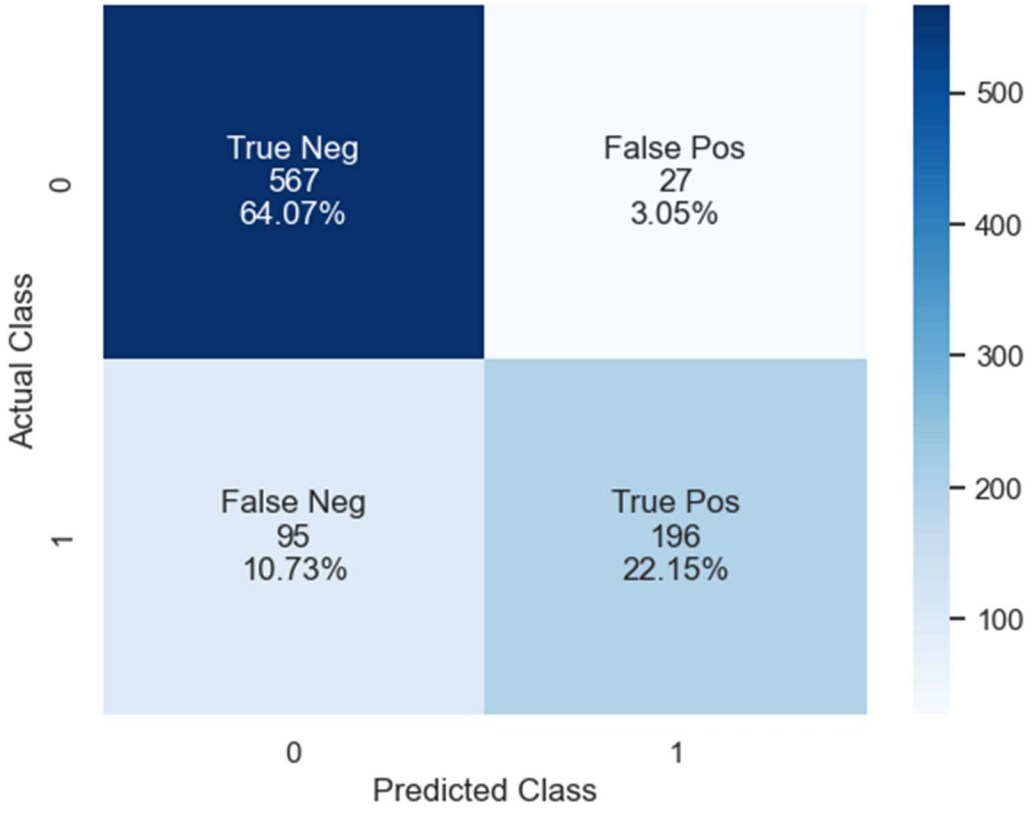


Training Accuracy continues to rise, indicating the model is learning from the training data and achieving near 0.89 accuracy. Validation Accuracy remains fairly stable around 0.86 but shows some fluctuations, hinting at mild overfitting. Training Loss steadily decreases, which is expected as the model optimizes. Validation Loss shows significant fluctuation and does not consistently decrease, indicating the model may be struggling to generalize well on unseen data.

Model 1 Performance (Test Set):

- Accuracy: 86%
- Precision: 87%
- Recall: 81%
- F1-Score: 83
- ROC-AUC: 91%

Confusion Matrix:



Model 2

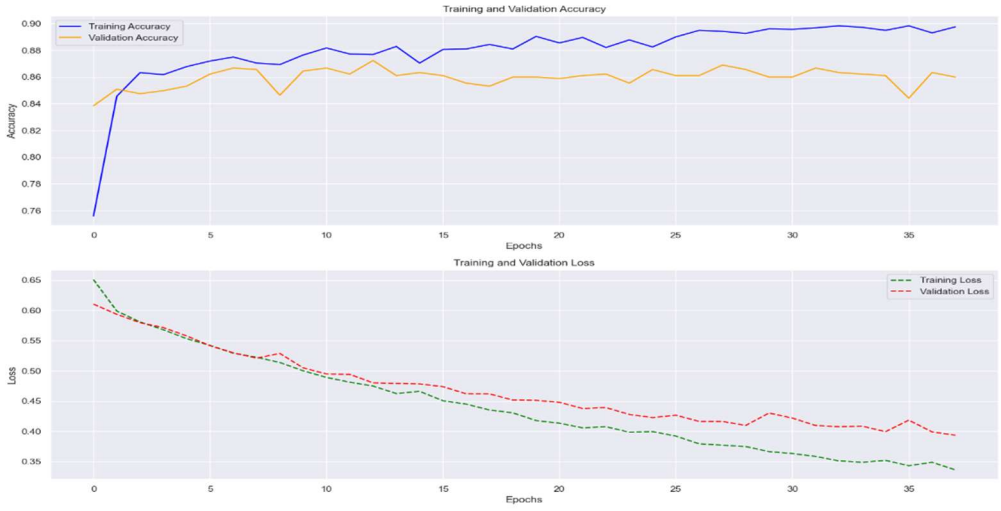
The model consists of the following layers:

1. **Input Layer**:
    - **Layer Type**: Dense (Fully Connected)
    - **Number of Neurons**: 128

- **Activation Function**: ReLU (Rectified Linear Unit)
- **Input Shape**: The input layer accepts a feature vector with a shape equal to the number of independent variables.

2. **Hidden Layers**:
   - **Layer 2**:
     - **Layer Type**: Dense
     - **Number of Neurons**: 64
     - **Activation Function**: ReLU
   - **Layer 3**:
     - **Layer Type**: Dense
     - **Number of Neurons**: 8
     - **Activation Function**: ReLU
   - **Output Layer**:
     - **Layer Type**: Dense
     - **Number of Neurons**: 1
     - **Activation Function**: Sigmoid

Learning Curve Analysis:

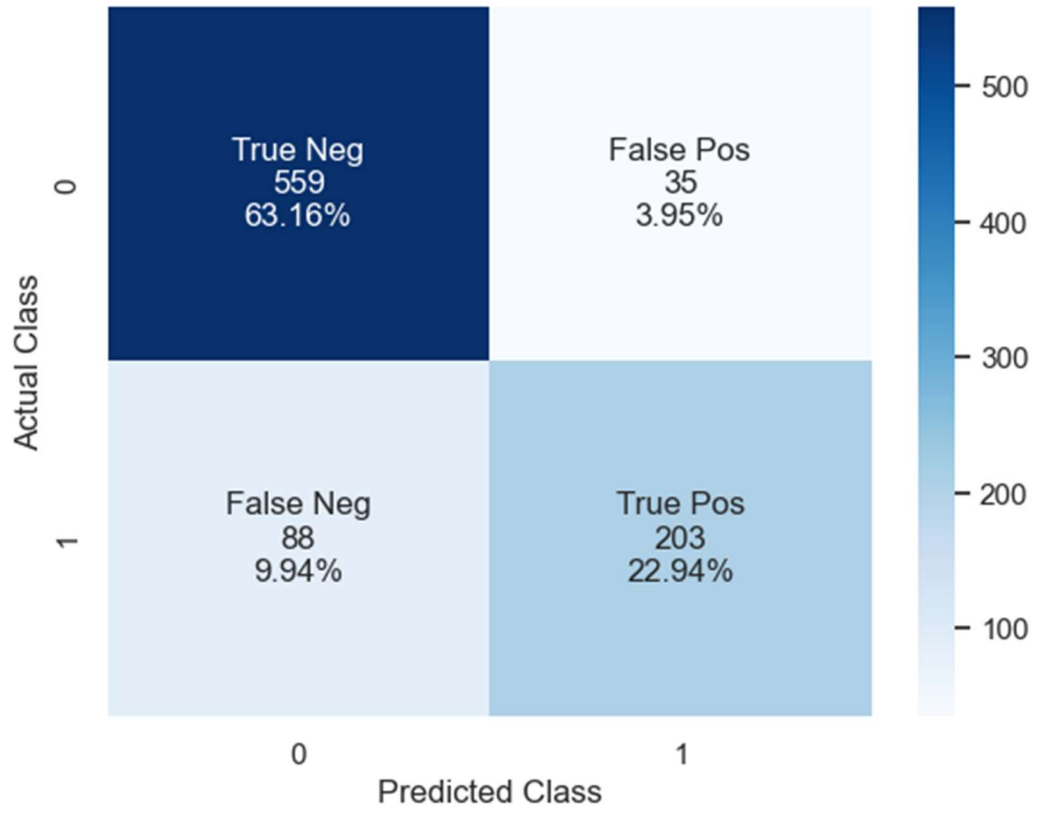


Training Accuracy continues to improve slightly and stabilizes around 0.88, indicating consistent learning. Validation Accuracy stabilizes around 0.85, with minimal fluctuations, suggesting good generalization. Training Loss steadily decreases, which reflects effective model learning on the training data. Validation Loss also decreases, albeit with some fluctuations, and trends similarly to the training loss, indicating the model is improving on unseen data without significant overfitting.

Model 2 Performance (Test Set):

- Accuracy: 86%
- Precision: 86%
- Recall: 82%
- F1-Score: 83
- ROC-AUC: 90%

Confusion Matrix:

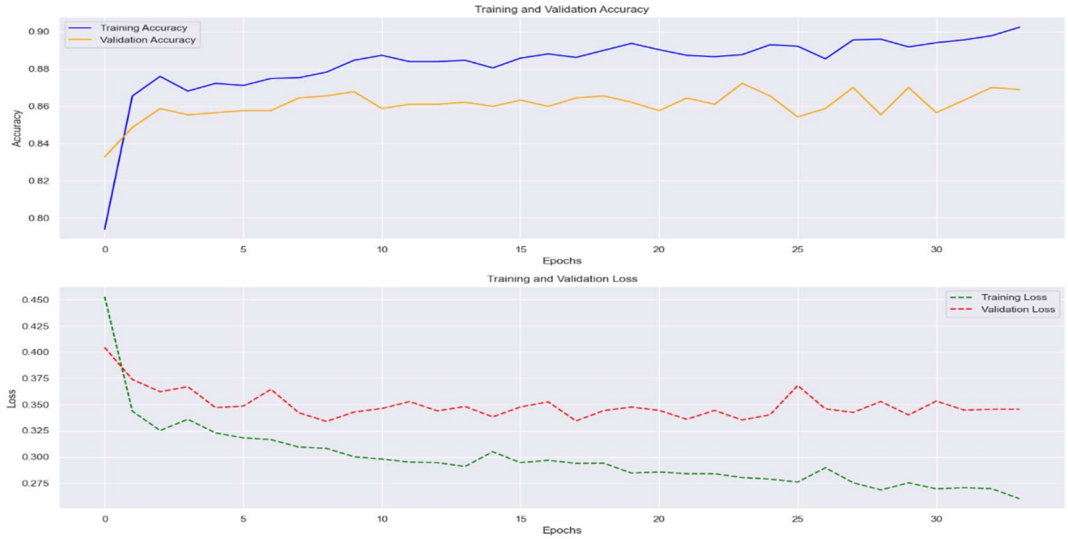


Model 3

The model consists of the following layers:

1. **Input Layer**:
    - **Layer Type**: Dense (Fully Connected)
    - **Number of Neurons**: 256
    - **Activation Function**: ReLU (Rectified Linear Unit)
    - **Input Shape**: The input layer accepts a feature vector with a shape equal to the number of independent variables.
2. **Hidden Layers**:
    - **Layer 2**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 128
        - **Activation Function**: ReLU
    - **Layer 3**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 8
        - **Activation Function**: ReLU
    - **Layer 4**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 8
        - **Activation Function**: ReLU
3. **Output Layer**:
    - **Layer Type**: Dense
    - **Number of Neurons**: 1
    - **Activation Function**: Sigmoid

Learning Curve Analysis:



Training Accuracy continues to improve slightly and stabilizes around 0.89, indicating consistent learning. Validation Accuracy stabilizes around 0.87, with minimal fluctuations, suggesting good generalization. Training Loss steadily decreases, which reflects effective model learning on the training data. Validation Loss also decreases, albeit with some fluctuations, and trends similarly to the training loss, indicating the model is improving on unseen data without significant overfitting.

Model 3 Performance (Test Set):

- Accuracy: 87%

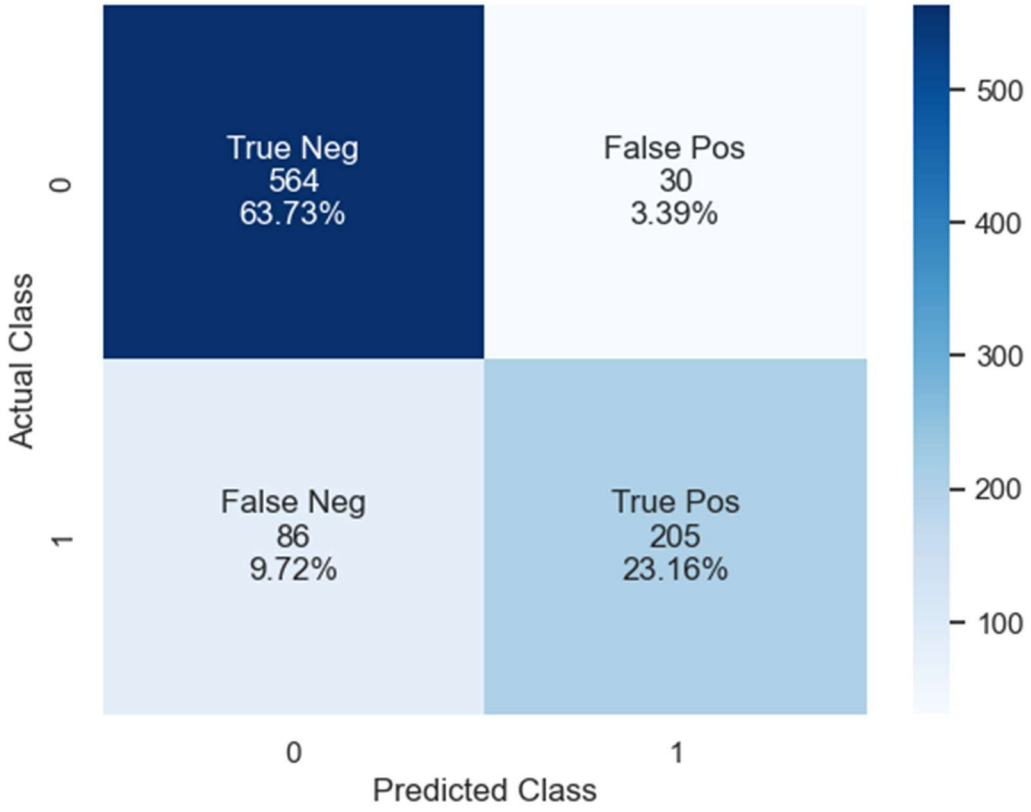
- Precision: 87%

- Recall: 83%

- F1-Score: 84

- ROC-AUC: 92%

Confusion Matrix:



Model 4

The model consists of the following layers:

1. **Input Layer**:
    - **Layer Type**: Dense (Fully Connected)
    - **Number of Neurons**: 2048
    - **Activation Function**: ReLU (Rectified Linear Unit)
    - **Input Shape**: The input layer accepts a feature vector with a shape equal to the number of independent variables.
2. **Hidden Layers**:
    - **Layer 2**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 1024

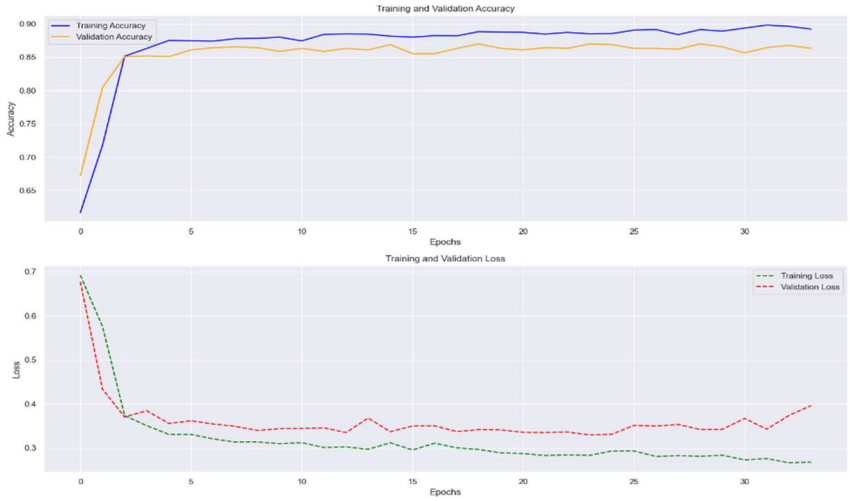
        - **Activation Function**: ReLU
    - **Layer 3**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 512
        - **Activation Function**: ReLU
    - **Layer 4**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 128
        - **Activation Function**: ReLU
    - **Layer 5**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 64
        - **Activation Function**: ReLU
    - **Layer 6**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 32
        - **Activation Function**: ReLU
    - **Layer 7**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 16
        - **Activation Function**: ReLU

    - **Layer 8**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 8

- **Activation Function**: ReLU

3. **Output Layer**:
    - **Layer Type**: Dense
    - **Number of Neurons**: 1
    - **Activation Function**: Sigmoid

Learning Curve Analysis:



- **Training and validation accuracy:** Both the training and validation accuracy are relatively high, suggesting that the model is not underfitting. However, there is a noticeable gap between the two curves, indicating that the model might be overfitting to some extent.
- **Training and validation loss:** The training loss decreases steadily, while the validation loss starts decreasing but then plateaus or even starts increasing. This is another sign of overfitting, as the model is learning the noise in the training data but not generalizing well to unseen data.

Model 4 Performance (Test Set):

- Accuracy: 86%

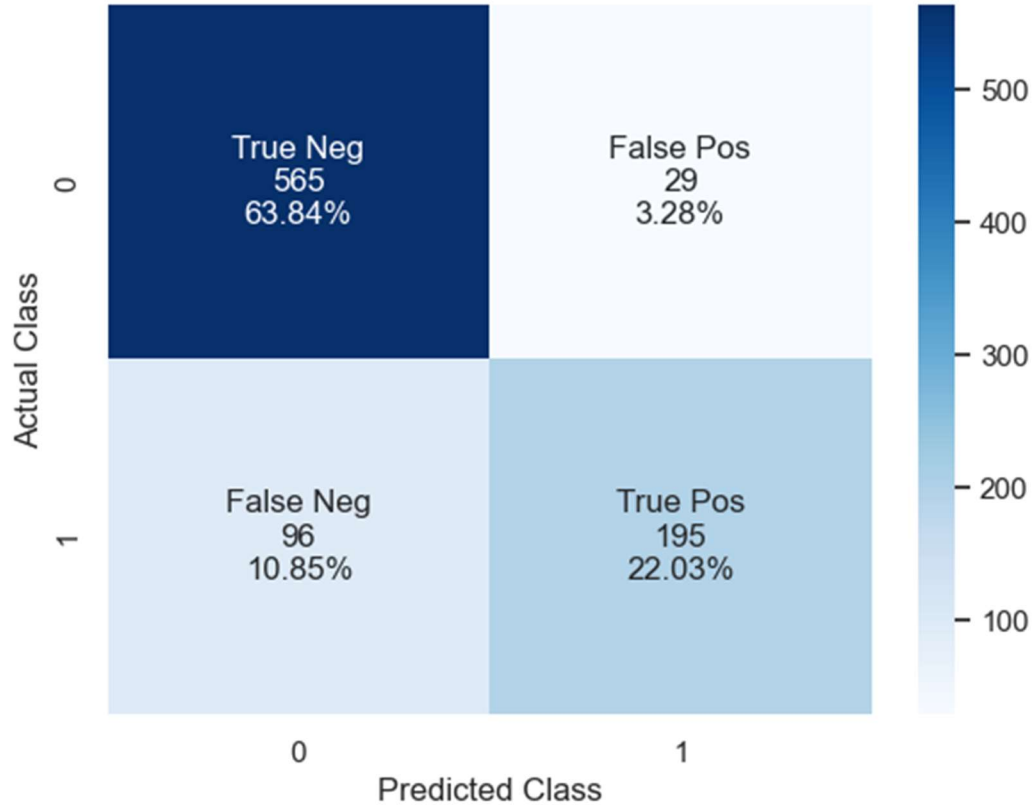
- Precision: 86%

- Recall: 81%

- F1-Score: 83

- ROC-AUC: 92%

Confusion Matrix:



Model 5

The model consists of the following layers:

1. **Input Layer**:
    - **Layer Type**: Dense (Fully Connected)
    - **Number of Neurons**: 4096
    - **Activation Function**: ReLU (Rectified Linear Unit)
    - **Input Shape**: The input layer accepts a feature vector with a shape equal to the number of independent variables.
2. **Hidden Layers**:
    - **Layer 1**:
        - **Layer Type**: Dense

        - **Number of Neurons**: 2048
        - **Activation Function**: ReLU
        - **Regularization**: Dropout - 0.5

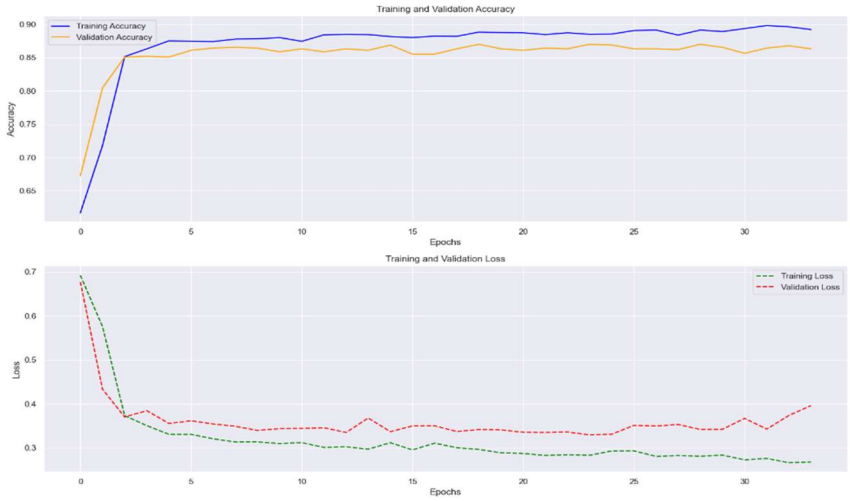
    - **Layer 2**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 1024
        - **Activation Function**: ReLU
        - **Regularization**: Dropout - 0.5
    - **Layer 3**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 512
        - **Activation Function**: ReLU
    - **Layer 4**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 128
        - **Activation Function**: ReLU
    - **Layer 5**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 64
        - **Activation Function**: ReLU
    - **Layer 6**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 32
        - **Activation Function**: ReLU

- **Layer 7**:
  - **Layer Type**: Dense
  - **Number of Neurons**: 16
  - **Activation Function**: ReLU

- **Layer 8**:
  - **Layer Type**: Dense
  - **Number of Neurons**: 8
  - **Activation Function**: ReLU

3. **Output Layer**:
   - **Layer Type**: Dense
   - **Number of Neurons**: 1
   - **Activation Function**: Sigmoid

Learning Curve Analysis:



- **Training and validation accuracy**: Both the training and validation accuracy are relatively high, suggesting that the model is not underfitting. However, there is a noticeable gap between the two curves, indicating that the model might be overfitting to some extent.

- **Training and validation loss:** The training loss decreases steadily, while the validation loss starts decreasing but then plateaus or even starts increasing. This is another sign of overfitting, as the model is learning the noise in the training data but not generalizing well to unseen data.

Model 5 Performance (Test Set):

- Accuracy: 87%

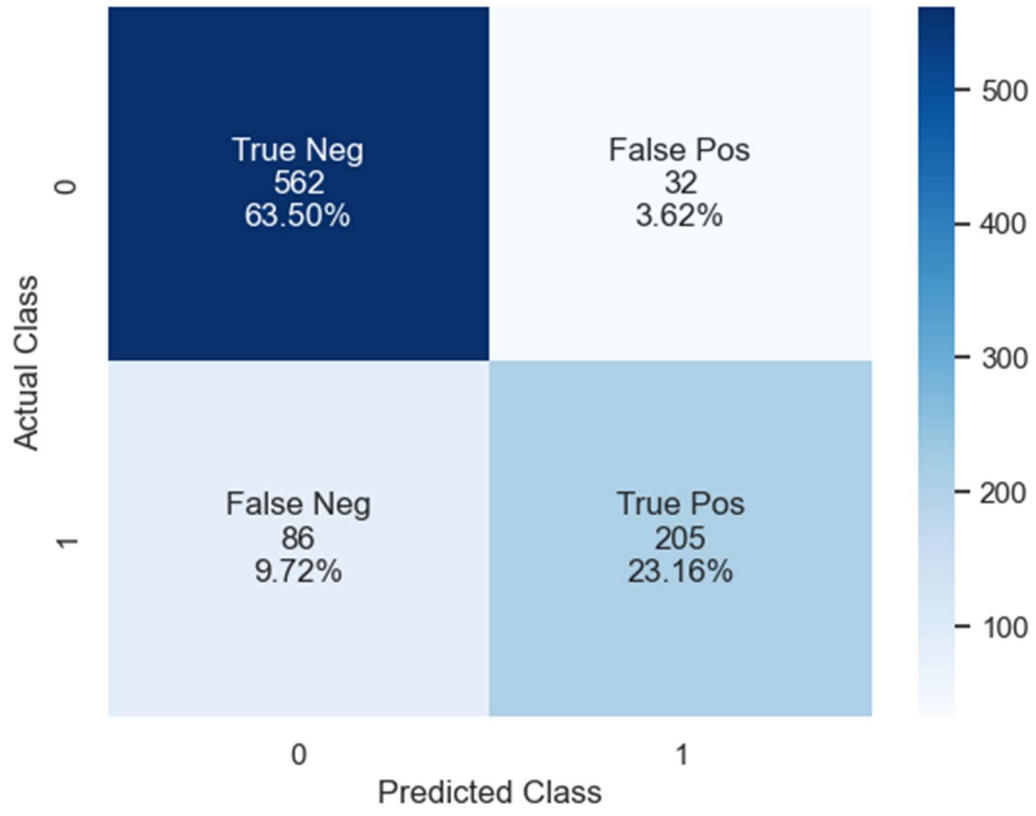
- Precision: 87%

- Recall: 83%

- F1-Score: 84

- ROC-AUC: 92%

Confusion Matrix:



Model 6

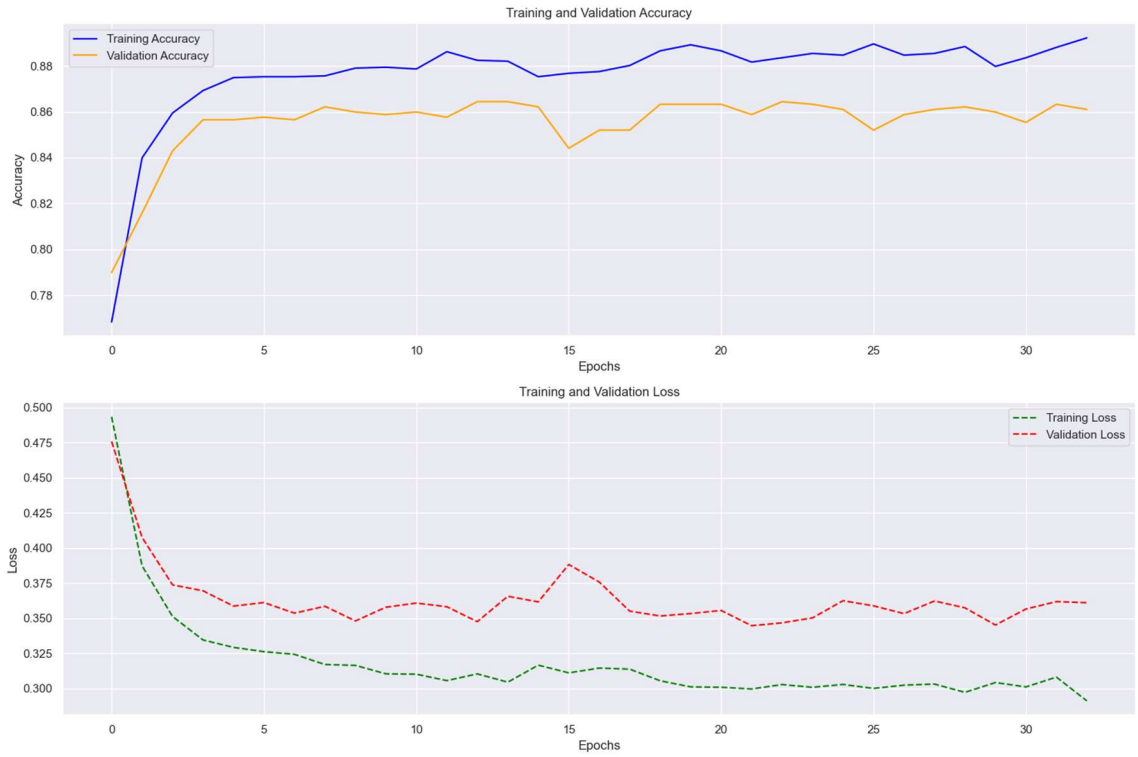
The model consists of the following layers:

1. **Input Layer**:
    - **Layer Type**: Dense (Fully Connected)
    - **Number of Neurons**: 4
    - **Activation Function**: ReLU (Rectified Linear Unit)
    - **Input Shape**: The input layer accepts a feature vector with a shape equal to the number of independent variables.
2. **Hidden Layers**:
    - **Layer 2**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 16
        - **Activation Function**: ReLU
    - **Layer 3**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 16
        - **Activation Function**: ReLU
    - **Layer 4**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 64
        - **Activation Function**: ReLU
    - **Layer 5**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 256
        - **Activation Function**: ReLU
    - **Layer 6**:
        - **Layer Type**: Dense
        - **Number of Neurons**: 1024

- **Activation Function**: ReLU

3. **Output Layer**:
    - **Layer Type**: Dense
    - **Number of Neurons**: 1
    - **Activation Function**: Sigmoid

Learning Curve Analysis:



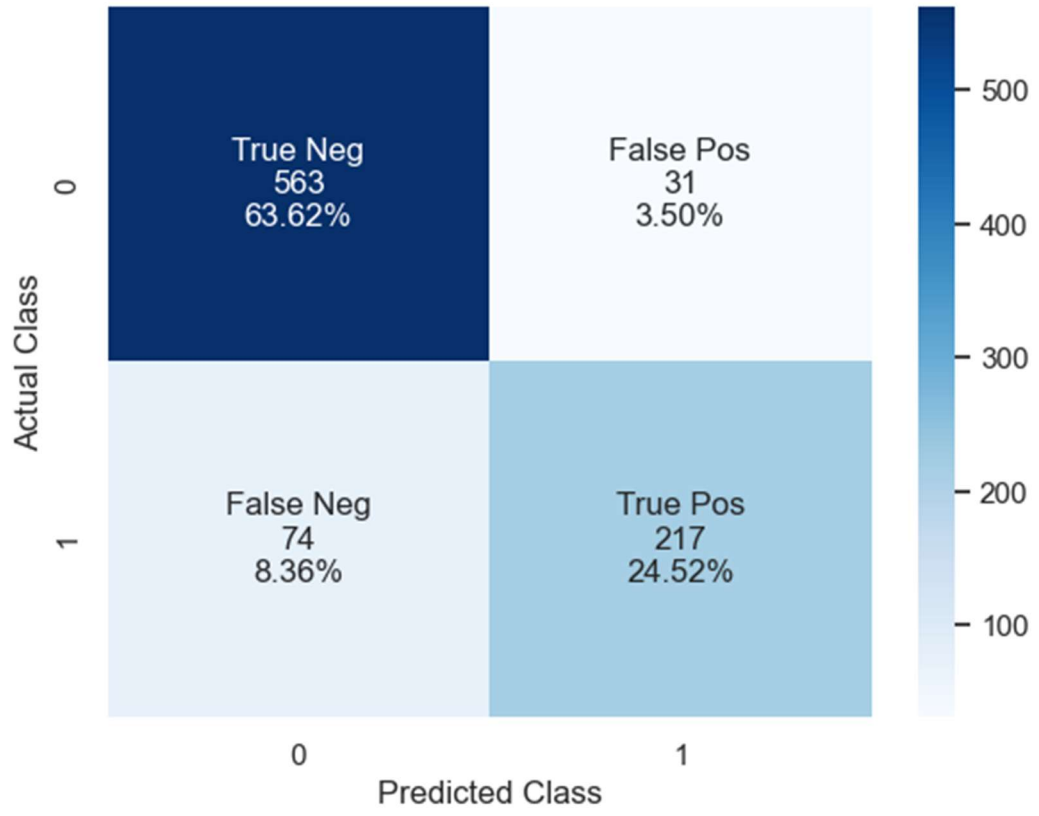
- **Training and validation accuracy:** Both the training and validation accuracy are relatively high, suggesting that the model is not underfitting. However, there is a noticeable gap between the two curves, indicating that the model might be overfitting to some extent.
- **Training and validation loss:** The training loss decreases steadily, while the validation loss starts decreasing but then plateaus or even starts increasing. This is another sign of overfitting, as the model is learning the noise in the training data but not generalizing well to unseen data.

Model 6 Performance (Test Set):

- Accuracy: 88%

- Precision: 88%
- Recall: 85%
- F1-Score: 86
- ROC-AUC: 91%

Confusion Matrix:



4. Model Evaluation and Comparison

4.1 Performance Comparison

The models were compared based on several performance metrics:

Traditional Machine Learning

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	87	87	83	85	92
Decision Trees	83	81	79	80	85
Random Forest	85	85	80	82	90
SVM	87	87	83	84	90
Gradient Boost	86	86	82	83	91
CatBoost	87	87	82	84	92
XGBoost	87	86	83	84	92

Logistic Regression and SVM (Support Vector Machine) models demonstrate the highest accuracy at 87%, indicating strong predictive capabilities.Both models also exhibit high precision (87%) and recall (83%), resulting in robust F1-scores (85% for Logistic Regression and 84% for SVM). Additionally, their ROC-AUC values are notably high, with Logistic Regression at 92% and SVM at 90%, suggesting excellent performance in distinguishing between classes.

CatBoost and XGBoost models also achieve 87% accuracy, matching the top performers. However, they show a slight decrease in recall (82% for CatBoost and 83% for XGBoost) compared to Logistic Regression and SVM. Their F1-scores are 84%, and ROC-AUC values stand at 92%, indicating strong overall performance.

Gradient Boosting and Random Forest models follow closely with accuracies of 86% and 85%, respectively. Their precision and recall metrics are slightly lower, leading to F1-scores of 83% for Gradient Boosting and 82% for Random Forest. ROC-AUC values are 91% for Gradient Boosting and 90% for Random Forest, reflecting their effectiveness in classification tasks.

Decision Trees exhibit the lowest performance among the listed models, with an accuracy of 83%, precision of 81%, recall of 79%, and an F1-score of 80%. The ROC-AUC value is 85%, indicating a moderate ability to distinguish between classes.

Deep Learning

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Initial ANN	87	86	83	84	91
Model 1	86	87	81	83	91
Model 2	86	86	82	83	90
Model 3	87	87	83	84	92
Model 4	86	86	81	83	92
Model 5	87	87	83	84	92
Model 6	88	88	85	86	91

Model 6 demonstrates the best overall performance with the highest accuracy (88%), precision (88%), recall (85%), and F1-score (86%). Its ROC-AUC score of 91% indicates strong classification ability. Model 3, Model 5, and the Initial ANN also perform well, with 87% accuracy, 87% precision, 83% recall, and an F1-score of 84%. Their ROC-AUC scores are 91%-92%, indicating solid classification performance.

Model 1 and Model 2 have slightly lower recall values (81%-82%), resulting in marginally lower F1-scores (83%). Despite this, their precision and ROC-AUC scores remain competitive, suggesting they still perform well overall.

Model 4 shows similar performance to Model 1 and Model 2, with a high ROC-AUC score of 92% but slightly lower precision and recall.

Models like Logistic Regression and Decision Trees are highly interpretable, making them suitable when transparency is essential, such as in regulated industries. Random Forest offers some interpretability through feature importance, while models like XGBoost, Gradient Boost, and CatBoost are harder to interpret but offer high performance.

All the deep learning models are variants of Artificial Neural Networks, which are inherently less interpretable. While they may provide strong predictive performance, understanding why specific predictions are made is much more challenging. These models are suited for scenarios where interpretability is less critical, and the focus is more on maximizing predictive power.

5. Recommendations for Model Selection

The recommended model is the Model 6 which is a deep learning variant, as it offers higher accuracy and precision. The logistic regression model is the next best as it offers both good performance and interpretability.