Contents

MODEL DEVELOPMENT REPORT

Details of Model Architectures

1.1. Logistic Regression

Is a statistical method used for binary classification where the outcome variable is categorical with two outcome values which could be either be a success or failure. It is a linear model that estimates probabilities using the logistic sigmoid function, which transforms it between 0 and

1.2. Decision Tree

It is a non-parametric supervised learning algorithm, used for classification and regression problem. They represent decisions and consequences in a tree like structure.it splits data based on feature values to make predictions.

A Decision tree consists of nodes (features), branches (rules), leaves(outcome) the root node represents the entire dataset.

The decision tree uses criteria like the Gini impurity to gain information on how to split the data

1.3. Random Forest

An ensemble learning method that builds multiple decision trees and merges their results to improve Accuracy and avoid overfitting.

Key Concept:

Ensemble: combines the prediction of multiple decision trees to increase robustness and accuracy

Bootstrapping: each tree is trained on random subset of the data allowing for diversity in the model.

Feature randomness: at each split in the tree only a random set of features is considered which helps reducing overfitting

1.4. Support Vector Machine

A classification algorithm that finds the optimal hyper plane to separate different classes of data. It is used for regression and classification tasks.

Key concepts:

Hyper plane: Is a decision boundary that separates different classes in the feature space, in two dimensional space it is a line and in 3 dimensional space it is a plane.

Support vectors: the data points that are close to the hyper plane and influence its position and orientation

Kernel trick: allows SVM to handle non-linear relationships by transforming the data input into higher dimensional plane. Examples of kernel include linear, polynomial and radial basis function

1.5. XGBoost

An efficient and scalable implementation of gradient boosting for supervised learning tasks. It is an ensemble techniques that combine weak learners by iteratively improving on the errors made in the previous models. It can effectively manage dataset with missing values by learning which path to take based on absence or presence of data.

1.6. Neural Network (TensorFlow)

A deep learning architecture composed of layers of neurons, using backpropagation to adjust weights during training. The TensorFlow model consists of fully connected neural network layers. These layers utilize for multiclass classification. ReLU activations in hidden layers and a softmax output layer.

Regularization techniques like Dropout may be applied to prevent overfitting, and optimization is typically done using Adam or SGD optimizers.

2.1. Comparison of model performance with visualization

Table 1: Comparison of model performance

model	Logistic Regression	Decision tree	Random forest	SVM	XGboost	CNN	Remark
Accuracy	Training:0.81 Testing:0.76	Training:0.1 Testing:0.65	Training:0.1 Testing:0.75	Training:0.83 Testing:0.73	Training:0.1 Testing:0.74	Model 1 Training:0.9929 Testing:0.712 Model 2 Training:0.8274 Testing:0.7522 Model 3 Training:0.785 Testing:0.7405 Model 4 Testing:1.0 Training:0.7260 Model 5 Training:0.7751 Testing:0.7474	The accuracy of the model on the training dataset across models is high but reduces on the test dataset, indicating overfitting, the larger the difference between the train and test accuracy score the higher the overfitting. The overfitting in model 5 is low.
Precision Score	Training:0.764 Testing:0.699	Training:1.0 Testing:0.584	Training:1.0 Testing:0.697	Training:0.843 Testing:0.685	Training:1.0 Testing:0.68		
F1_score	Training:0.715 Testing:0.583	Training:1.0 Testing: 0.583	Training:1.0 Testing: 0.658	Training:0.753 Testing:0.617	Training:1.0 Testing:0.663		
Recall_score	Training:0.704 Testing:0.666	Training:1.0 Testing:0.583	Training:1.0 Testing:0.653	Training:0.730 Testing:0.6157	Training:1.0 Testing:0.656		

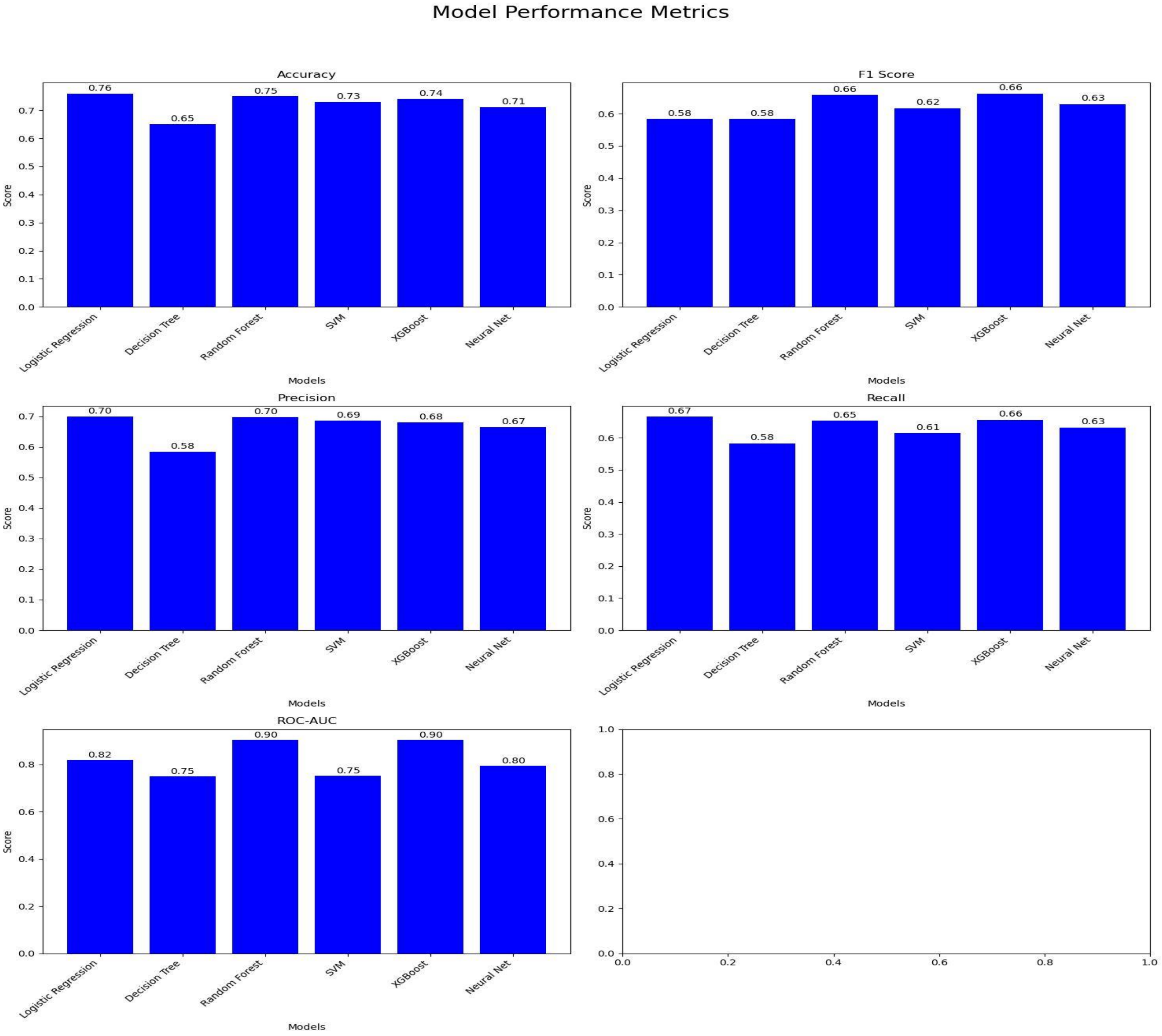

Figure 1: Visualization of model metrics

Logistic Regression:

- **Strengths**: Provides balanced performance between precision and recall, with an accuracy of 76%. Suitable when transparency and simplicity are needed.
- **Weaknesses**: Lower F1-score compared to Random Forest and XGBoost, which indicates that it may miss some positive instances.

Decision Tree:

- **Strengths**: Simple to interpret, but suffers from overfitting, which is reflected in its low testing accuracy (65%).

- **Weaknesses**: Poor generalization, as evident by a significant drop in performance metrics like F1-score and precision.

Random Forest:

- **Strengths**: Excellent performance across all metrics, with a high ROC-AUC of 0.9035, making it one of the top-performing models. The random forest handles both precision and recall very well.
- **Weaknesses**: Slight overfitting but still a robust model.

Support Vector Machine (SVM):

- **Strengths**: Performs reasonably well in some metrics like recall but underperforms in terms of precision and F1-score.
- **Weaknesses**: Its ROC-AUC is lower compared to Random Forest and XGBoost, which suggests it's not distinguishing between classes as effectively.

XGBoost:

- **Strengths**: Top performance in terms of both ROC-AUC and accuracy (0.9039 ROC-AUC and 74% accuracy). Excellent generalization across all metrics.
- **Weaknesses**: Slight complexity in tuning, but overall the strongest performer.

Neural Network (TensorFlow):

- **Strengths**: Decent performance across all metrics with a ROC-AUC of 0.795. The neural network architecture allows it to perform well on larger datasets.
- **Weaknesses**: Slightly lower precision and F1-score compared to XGBoost and Random Forest, indicating some room for optimization.

Applying cross-validation

Cross validation is a techniques applied to a model to improve the performance and reduce overfitting, by training and testing it on multiple subsets. The dataset is divided into fold, each fold act as the training and validation data. To implement cross validation the dataset is split into train and test, the train is further split into k subsets, the model train on k-1 subset and evaluate with the remaining subset, then calculate the average performance of the metric.

The table below summarizes model performance metrics across all models (including TensorFlow). These metrics were computed using 5-fold cross-validation:

Table 2: Comparison of model performance

	Logistic Regression	Decision Tree	Random Forest	SVM	XGboost	CNN
Precision	0.6654	0.6799	0.7600	0.2498	0.7690	
Recall score	0.6838	0.6804	0.7744	0.4993	0.7780	
F1 score	0.6327	0.6798	07582	0.3329	0.7696	
accuracy	0.6838	0.6804	0.7744	0.4993	0.7780	0.4993
ROC-AUC	0.8199	0.7496	0.9035	0.7531	0.9039	

3.1. Analysis of learning curve and model Diagnostics

The learning curves provide key insights into how models are generalizing as training progresses:

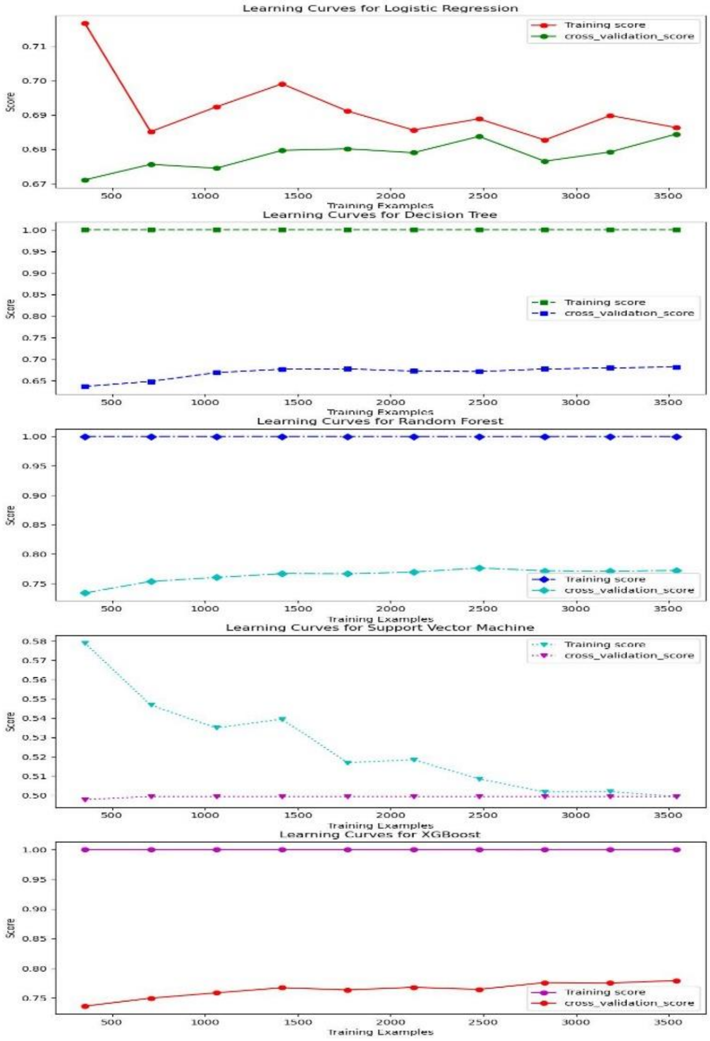

Figure 2: learning curve for machine learning models

1. Logistic Regression: The training and validation curves converge, indicating a balanced model with minimal overfitting.

2. Decision Tree: The gap between training and validation curves signals significant overfitting. More data or regularization could help.

3. Random Forest: The model demonstrates strong performance with curves closely aligned.

4. Support Vector Machine: While the SVM shows promise, there is potential underfitting with less alignment between training and validation.

5. XGBoost: The learning curve shows XGBoost excelling with a smooth and close gap between training and validation, suggesting it is a top performer.

6. TensorFlow Neural Network: Initially, the TensorFlow model's training curve may show rapi growth, but Dropout regularization keeps the validation curve stable. Some minor overfitting may still appear, but adjustments to batch size, learning rate, or network depth can improve generalization.

Learning curve for Tensorflow model

Model 1: it has 2 layers and the layer has 128 and 64 neurons respectively

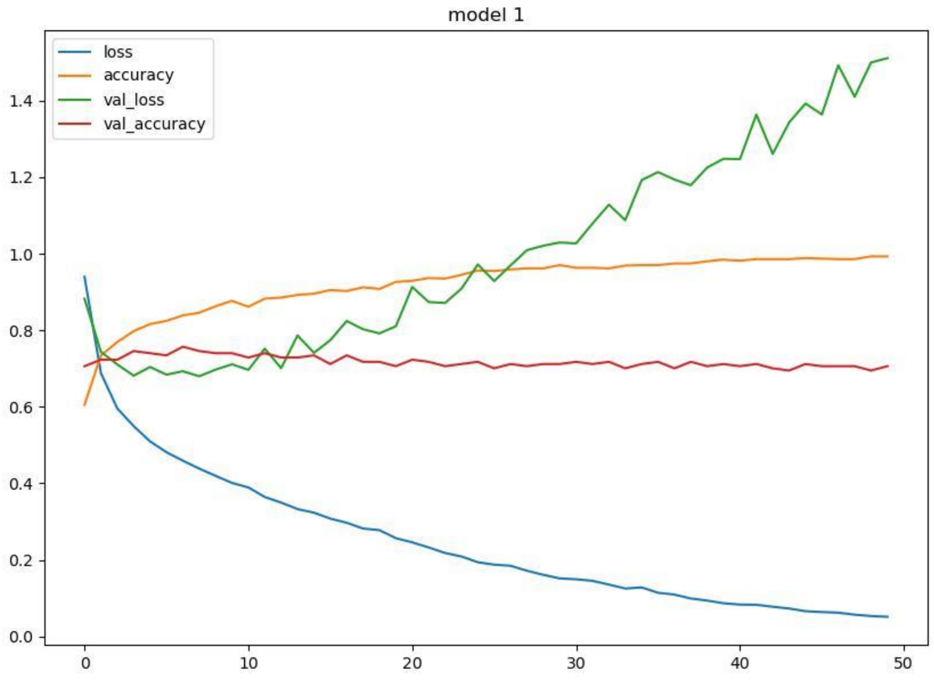

Figure 3: learning curve for tensorflow model 1

- Green Line: this represents the validation loss, an increase in the validation loss indicates that the model is not generalizing very well.
- Orange Line: this represent the train accuracy the line increases for a short period and remains stable, this indicates that the model is performing well on training dataset
- Red line: this represents the model performance accuracy on the validation dataset, the wide gap between the validation accuracy and training accuracy indicates the model is overfitting

- Blue line: this represents loss on the train dataset, the decrease indicates that the model is doing well on the train dataset.

Model 2: it has two layers and a dropout of 0.5 the layers have 64 and 32 neurons respectively

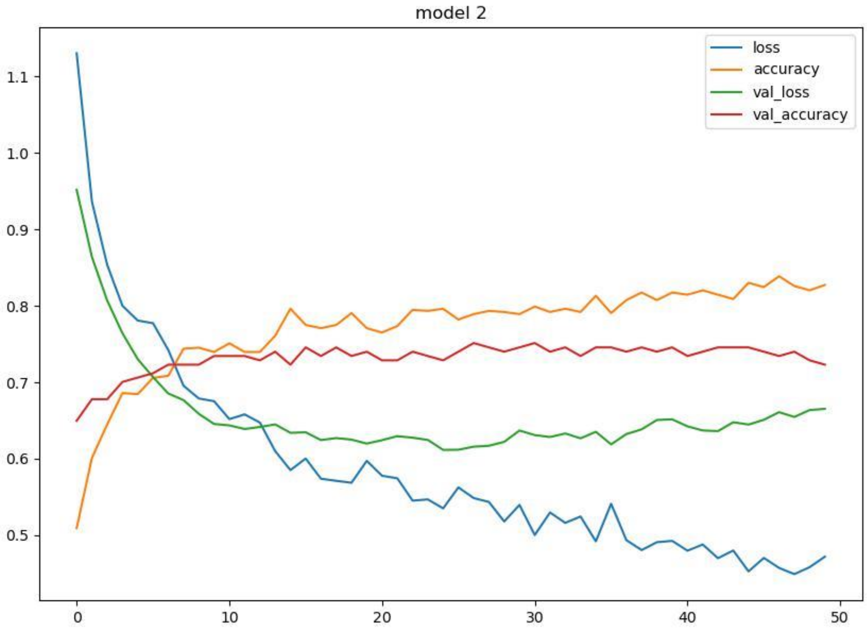

Figure 4: learning curve for Tensorflow model 2

- the green and blue line represents loss in validation and train respectively the fall in the curve indicates that there is low loss in the model performance on train and validation dataset
- the orange and red line represent the accuracy of the model on train and validation data respectively,
- the orange line is seen point up, a sign that the model is performing well on the train dataset, while the red line point downward indicating that if trained for more epochs the validation accuracy can drop, the small gap between the orange and red line indicates there is reduction in the overfitting as compared to model 1

Model 3: it has 3 layers with a dropout of 0.3 each layers has neurons of 64, 32 and respectively

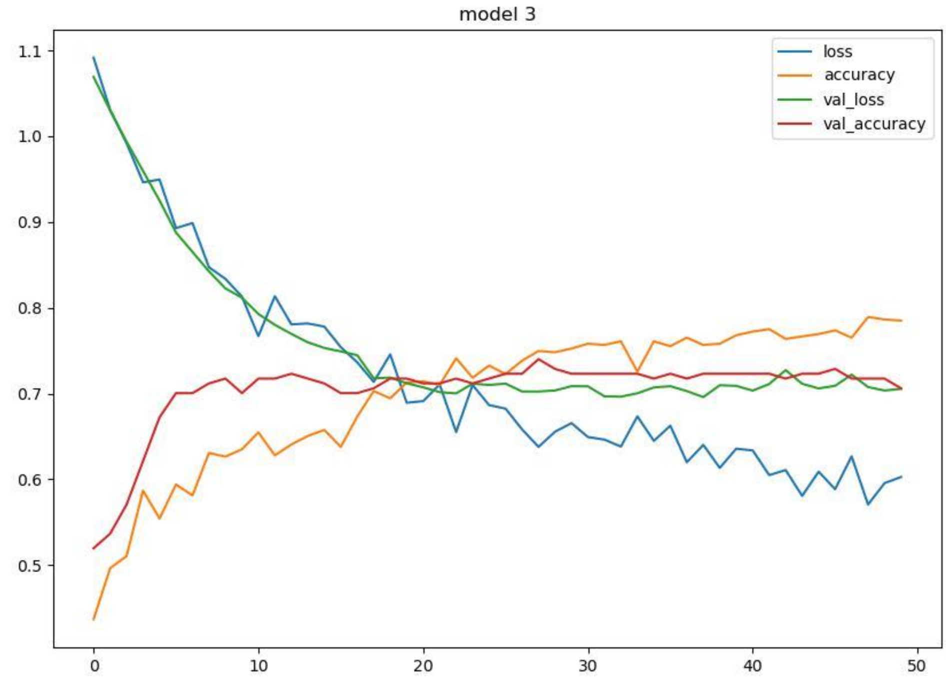

Figure 4: learning curve for tensorflow model 3

- the green and blue line represents loss in the validation and train dataset, the blue line points up indicating that the train can still increase, the validation loss is very close to validation accuracy(red line),
- there is a reduction in the distance between the validation and train accuracy indicating a further reduction in overfitting
- the validation accuracy seems to point downward indicating a possibility in the drop in performance if training continues

Model 4: it has 4 layers 128, 64, 32 and 8 respectively

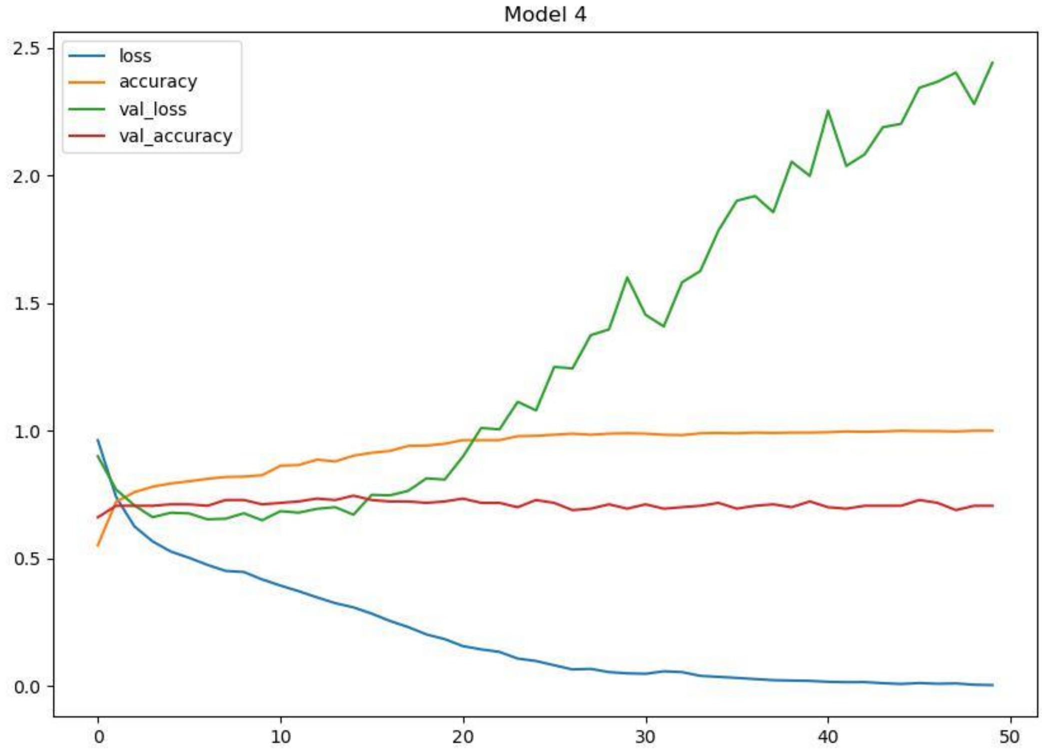

Figure 5: learning curve for tensoflow model 4

- the increase in the validation loss(green line) indicates that this model is not generalizing well
- the drop in the train loss (blue line) indicates that the model is performing well on train data
- The train accuracy and validation accuracy is straight which means the model is performing well.
- The gap between the validation and train accuracy indicates the presence of overfitting

Model 5: it has 3 layers and dropout of 0.3, which have 64, 32 and 8 neurons respectively

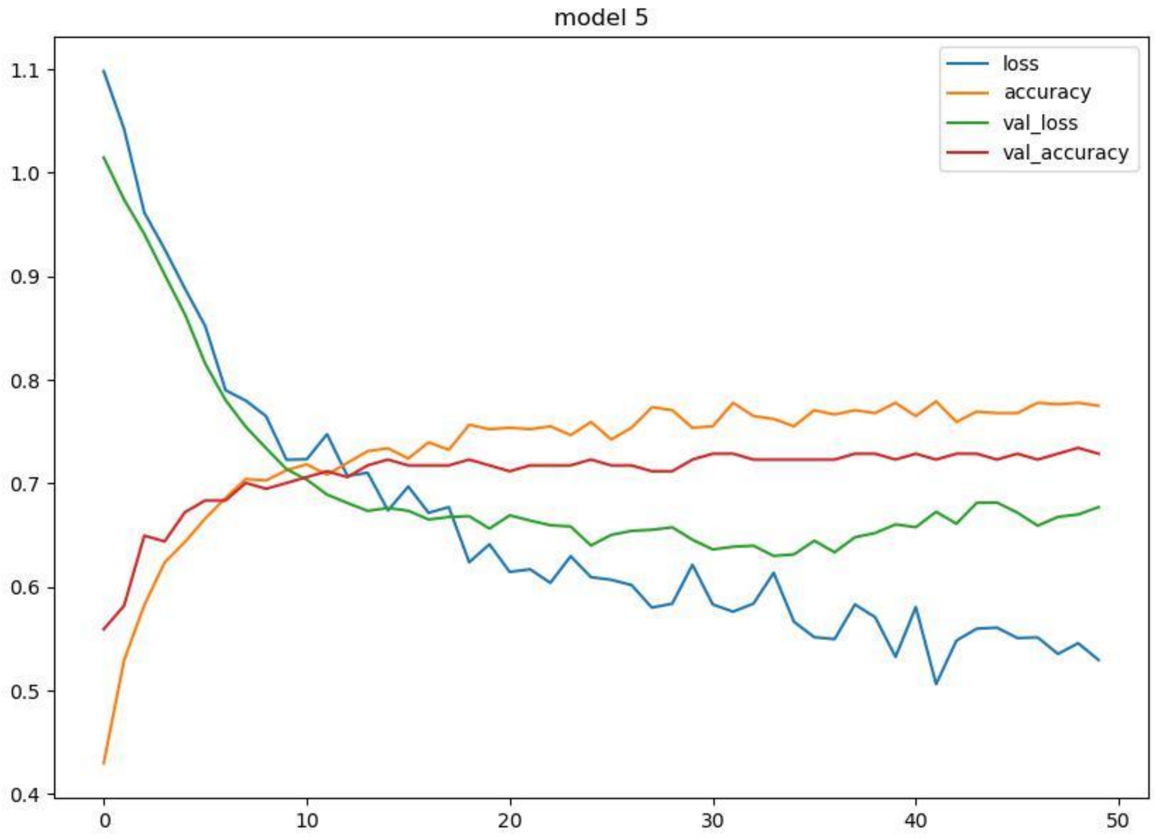

Figure 6: Learning curve for tensorflow model 5

- The train loss(blue line) dives downward and seems to continue in that direction
- The validation loss(green line) dives down and begin to move upward
- The validation and train accuracy is have a small gap between, which indicates a reasonable reduction in overfitting

Model Selection Recommendations

Recommendations for Model Selection:

1. For high performance: XGBoost continues to be the strongest contender, achieving both high accuracy and ROC-AUC scores.
2. For balanced performance and interpretability: Logistic Regression remains an excellent choice for transparency.
3. For deep learning: The TensorFlow model, while powerful, is slightly more complex to tune. It is ideal when working with larger datasets or where deep feature learning is critical.
4. For simplicity: Random Forest offers strong performance without requiring significant hyperparameter tuning, making it a great out-of-the-box solution.

The choice between TensorFlow (neural network) and XGBoost may depend on dataset size and complexity: TensorFlow thrives on larger datasets with complex relationships, while XGBoost delivers fast, scalable performance.

Githublink: https://github.com/Insight00001/week4task.git