Checkpoint 3

BFI-Sustainability Team

Team Members - Akash Kesani, Ravindu Tharanga Perera, Samuel Kannan

Justification of Model Choice

We chose a neural network model which is a fully connected, or dense, neural network, often referred to as a multilayer perceptron (MLP).
Here's a breakdown of its components and architecture:

1. Input Layer:
    - The input layer has as many neurons as there are features in the dataset. In your case, it has three neurons corresponding to the three predictors: "Priority," "Flag Subcategory ID," and "Organization ID".
2. Hidden Layers:
    - The model contains two hidden layers. Each hidden layer has 64 neurons, which is a common choice for moderately complex problems. The more neurons and layers, the more capable the network is of learning complex patterns. However, too many can lead to overfitting.
    - Both hidden layers use the ReLU (Rectified Linear Unit) activation function. ReLU was chosen for its efficiency and effectiveness in non-linear transformations without suffering from vanishing gradients in deeper networks.
3. Output Layer:
    - The output layer's number of neurons corresponds to the number of unique classes in the "Status" target variable. This is typical for classification tasks where each neuron represents a class probability.
    - The softmax activation function is used in the output layer. Softmax is suitable for multi-class classification tasks as it outputs the probabilities of each class, with the total summing to 1.
4. Loss Function:
    - The model uses the categorical cross entropy loss function, which is standard for multi-class classification problems. It measures the difference between the predicted probabilities and the true values, effectively guiding the training process.

5. Optimizer:
    - The Adam optimizer is used, which is an extension of the stochastic gradient descent method. It is favored for its fast convergence and automatic tuning of the learning rate during training.

This setup with neural networks was adept in solving the classification task regarding the ‘Status’ response variable, providing a good balance between model complexity and performance. The chosen architecture and methods are robust for a wide range of problems if compared to our model.

Analyzing the relationship between response variable and predictors:

To analyze the relationship between the predictors ("Priority", "Flag Subcategory ID", and "Organization ID") and the outcome variable ("Status"), we perform statistical tests that evaluate the association between these variables. Given that both the predictors and the outcome variable are categorical or can be treated as such, Chi-square tests of independence are a suitable choice. The Chi-square test will help determine if there's a significant association between each predictor and the outcome.

The results from the Chi-square tests of independence between each predictor and the outcome ("Status") show the following:

1. Priority:
    - Chi-square Statistic: 125.89
    - P-value: $2.95\times10-26$
    - The very small p-value suggests a statistically significant association between "Priority" and "Status".
2. Flag Subcategory ID:
    - Chi-square Statistic: 4720.63
    - P-value: 0.00 (practically zero)
    - This result indicates a very strong association between "Flag Subcategory ID" and "Status"
3. Organization ID:
    - Chi-square Statistic: 1292.10
    - P-value: $1.94\times10-271$
    - Similarly, this very low p-value indicates a significant association between "Organization ID" and "Status".

These statistical tests suggest that all three predictors have significant associations with the outcome variable "Status". This indicates that changes in these predictors are likely to correspond to changes in the outcome, making them relevant choices for inclusion in our model.

Model’s Test Error Rate and Goodness of Fit

We used evaluation metrics such as confusion matrix, along with the precision, recall, and accuracy, that offer valuable insights into how well our model fits the data. Let's discuss each aspect:

Confusion Matrix:

```
 [[ 3313     0  5432]

 [   10    10   125]

 [ 2068     8 63755]]
```

- True Positive (TP) for the first class is 3313, which suggests that the model can correctly identify this class but also misses a substantial number (5432) by misclassifying them as the third class.
- Class 2 appears to be problematic, as the model fails to correctly predict any instances of this class (all are predicted as class 3). This indicates either an insufficient number of samples from this class in the training set or that the features do not differentiate this class well.
- Class 3 is predicted very well with a TP of 63755, but there are 2068 instances from class 1 and 2 misclassified as class 3, showing a strong bias towards predicting this class.

Precision

Precision for the model is approximately 0.883. Precision is the ratio of correctly predicted positive observations to the total predicted positives. High precision relates to a low false positive rate. A precision of 0.888 suggests that the model is quite reliable when it predicts a positive result. However, the very low number of samples for class 2 could be inflating the overall precision due to a low number of false positives caused by this class.

Recall

Recall (or sensitivity) of the model is 0.8977. This measure indicates that the model can identify 89.77% of all actual positives correctly. High recall indicates that the model is good at detecting the positives.

Accuracy

The accuracy of the model is 0.8977, which is generally quite high. This suggests that the overall number of correct predictions (both true positives and true negatives) is substantial. However, accuracy is not the best metric as it can be biased towards the more frequent classes.

Cases of Interest

In our data we find three interesting problems that can be solved by our neural network model. We then formulate the following three problems from our data and how our model solves the problem and compare it to our previous model based on logistic regression.

Large Dataset: Neural networks excel in handling large datasets. They have the capacity to learn complex patterns and relationships from a significant amount of data while our previous model did not perform well since the dataset was very large with complex nonlinear relationships, as they inherently assume a linear boundary between classes.

Imbalanced classes: Neural networks can be configured with techniques such as class weighting, specialized loss functions, or oversampling within the training process to handle class imbalance effectively while our previous model was still limited by the model's linear nature, and did not capture the complex boundaries between classes in an imbalanced dataset effectively.

Flexibility: Neural networks are highly flexible and capable of modeling complex and non-linear relationships due to their architecture (multiple layers, different activation functions). This makes them particularly suited for datasets where predictors interact in non-linear ways to affect the outcome while our previous model was limited to linear relationships which were manually extended but still did not capture very complex patterns.