**12 Initial Machine Learning Failures**

**Attempt 1: Using an Automated Neural Network Optimizer (Keras-Tuner)**

- We started by using an automated neural network optimizer called Keras Tuner. which can implement multiple layers, activation functions and nodes.

- It aims mimic the complexity of the human brain by connecting data points.

- We relied solely on our instincts to determine the most important features.

- The accuracy score was 0.559, which is quite low.

**12 Attempt 2: One-Hot Encoding the Dataset**

- In the second attempt, we decided to one-hot encode our dataset.

- One-hot encoding converts categorical data into binary values, allowing the algorithm to treat them as categories rather than numerical values.

- However, this approach can lead to problems when the number of categories becomes too high.

- In our case, we ended up with 117 columns after one-hot encoding, which introduced a lot of noise.

- The accuracy was even lower, with a score 0.32.

**13 Understanding Initial Machine Learning Failures**

* So we reduced data so we don’t have any outliers.
* Created a correlation map to identify columns with low correlation to the target variable

**14 PyCaret for Machine Learning**

* + Utilized PyCaret, an open-source, low-code machine learning library, for automated workflow
  + It utilizes a wide range of algorithms and techniques
  + PyCaret simplifies tasks such as data preparation, train-test split, and transformations
  + As you can see the table generated shows multiple models
  + Gradient Boosting Classifier was selected as the best model by PyCaret , with the highest accuracy score at 0.82
  + Tuned the model and achieved an accuracy score of 0.8228

**15 Leveraging Deep Learning for Improved Predictive Accuracy**

* + Returned to deep learning for improved predictive accuracy
  + Used the same dataset as in the previous PyCaret analysis
  + Utilized Keras Tuner to optimize the neural network model
  + Achieved a significant improvement in accuracy with a score of 0.86 much better.

**16 Summary of both models**

* Firstly, I want to go over some terms:
* Precision – is related to the positive observation so its TP/(TP+FP),
* so TP means that for example 1 = graduation, and the algorithm predicts that person will graduate and in the actual data they do graduate.
* False positive says if that the person will dropout (so classes it as a 0) but in the actual data it shows that they have graduated.
* Recall is the opposite and looks at the false negative. The calculation for that is TP/(TP+FP). So for example if we look at class 1, so graduate. The algorithm says that you are going to graduate but however in the actual data you dropped out. You want to mitigate away from that as much as possible.
* a recall of 0.94 for Class 1 means that the model correctly identified 94% of the actual instances that belonged to Class 1.
* The remaining 6% would be the false negatives, which means that the algorithm incorrectly predicted those instances as Class 1 when, in reality, they belonged to Class 0. This indicates instances that were misclassified by the model.
* Recall is a measure of the model's ability to capture all positive instances correctly. A lower recall value suggests that the model may have missed a significant portion of the instances that actually belonged to Class 1, leading to false negatives in the predictions.
* Precision: The Keras-Tuner model has higher precision values for both Class 0 (0.76) and Class 1 (0.89) compared to the PyCaret model (Class 0: 0.625, Class 1: 0.888). This means that the Keras-Tuner model had a higher proportion of correct positive predictions for both classes.
* Recall: The Keras-Tuner model has higher recall values for Class 1 (0.94) compared to the PyCaret model (0.878). However, the PyCaret model has a slightly higher recall for Class 0 (0.647) compared to the Keras-Tuner model (0.63).
* F1-score: The F1-scores for both models are relatively similar, with the Keras-Tuner model having a slightly higher F1-score for Class 1 (0.91) compared to the PyCaret model (0.883).
* Accuracy: The Keras-Tuner model has a higher accuracy (0.86) compared to the PyCaret model (0.8228).

Overall, the Keras-Tuner model tends to outperform the PyCaret model in terms of precision, recall (except for Class 0), and accuracy. However, it's important to consider other factors such as dataset characteristics, problem complexity, and specific requirements when evaluating and comparing models.