**2.**

Uncertainty always dominants choice-making or decision support. As a measure of uncertainty, entropy is the parameter that decides the nodes split in a decision tree, and the decision tree breaks a large dataset into smaller and smaller subsets until all the data classified into the same class. Each internal node is a test on a feature, and the information gain is based on the decrease in entropy regarding the bifurcation each time. An efficient system will be crucial for decreasing indeterminacy. Abundant real-world datasets include narrative columns, especially in the healthcare field. The decisive keyword retrieval is a valid way to filter the classification. This method could one-hot encode a narrative column into variables for a more accurate node split. A grading level measurement will promote to reduce uncertainty. Warning levels helping to distinguish personal situations will more effectively prompt information. Plotting the decision tree diagram is informative for visualization since the branch including sub-branches are conducive for people to understand the connections between two features. However, the more features in a large dataset, the more possible ways to split in the decision tree. This could augment the uncertainty. Data cleansing will be implemented into the system to erase redundant features and the original columns using to create the new variables. The system will be based on feature engineering to create features with identifying obviously wrong data and data assimilation to improve accuracies by continuously learning from data. The system adjusters will modulate features and parameters to reduce uncertainty.

Diagram:

A screenshot of a cell phone

Description automatically generated

Some narrative examples that support my system:

1. By definition, as untoward medical occurrences in a patient or clinical investigation subject, adverse events will be separated into the level from 1 to 5 according to the narratives of patients themselves. The system will be able to separate the population more effectively for further decision tree split and reduce uncertainties.
2. Decreasing the number of features will lower the uncertainties. To reduce the possible ways of split and uncertainties in the decision tree model to accelerate the computation, combining features is effective. The feature generation will smartly combine the features into a more conclusive and decisive feature to make predictions. In football, if a person’s running speed is fast and the striking level is high, we can merge the two columns into one to summarize this is a rush impact causing the injuries. At the same time, we erase the column of speed and striking level, and my decision tree will have fewer ways of split and uncertainties.
3. The real-world datasets will not be absolutely correct. In healthcare, not every patient will have enough knowledge to describe their circumstances accurately, and the physicians or data mangers are possibly typing wrong. The system managers will be able to use the feature engineering system to identify the wrong data, like a tennis ball past the net and drop in the court will definitely be counted as a win for the previous player’s hit. However, in some computer-recorded datasets, it will be counted as a loss for the previous player for no reason. The wrong data will increase uncertainty and negatively affect our prediction model.
4. Extracting keywords from a long narrative example is the technique we have learned. We could flexibly create binary columns to help our decision tree model to classify outcomes more efficiently with saving some computation resources. In the healthcare field, the system could customize each patient’s personal case condition by using the dummy variables we created more quickly.
5. Not only the one-hot encoding technique is commonly used in healthcare injuries with patients’ descriptions, but the technique is heavily used in risk management in loans or banking companies. They have to record key information as word narratives to evaluate the risk score for each borrower. The system erases some narratives with words and uses binary numbers will reduce uncertainties.

**3. Nodes analysis:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Head Injury Classification** | | **my\_def (True condition)** | |
| **1** | **0** |
| **my\_test** | **1** | **TP** | **FP** |
| **(Predicted condition)** | **0** | **FN** | **TN** |
|  |  | **Sensitivity = TP/(TP+FN)** | **Specificity = TN/(TN+FP)** |

I will use this table for my below analysis.

The five nodes I choose to analyze are node 36, node 32, node 28, node 8, and node 7.

1. **Node 36**

Node 36 is one of the topmost nodes in the decision tree since it has 17 leaves under this node. 17 terminal-node leaves mean there are possible some splits follow under the node 36. It includes the “LOC to be 0, SCALP to be 0, HEADS to be 0, HD to be 0, HEADACHE to be 0, COLLISION to be 0, CHI to be 0, CONCUSSION to be 0, and HEAD to be 0”. In node 36, there are 51 rows of data (0.68%) are classified as cases and 7,477 (99.32%) rows are classified as non-cases.

Node 36 uses “AND” to connect 9 keywords. When all of the above keywords are missing simultaneously, the prediction result for head injuries will be “0”. Otherwise, the prediction will be a “one”. This is a very high-level and general large condition because there are very fewer narratives including all of the above keywords exactly. This will not be a very effective classification. However, this node does help to correctly classify the True positives, which will be sensitivity. It will be effective for the dataset to predict the true positives since the condition is very strict to satisfy. Once a case row data is satisfied (all the keywords included), it must be a head injury. To improve the identity case, we can apply “proc sql” with clause WHERE my\_def = 1 and my\_test = 0 to show all the true cases that have not been identified, which are the false negatives. We can try to observe what the keyword is missing. On the other hand, we can print the dataset for WHERE my\_def = 0 and my\_test = 1 in “proc sql” to show the keywords for false positives to improve the specificity. In this case, 99.31 percent are classified as 0, which means there is an immense number of false negatives in our classification. Therefore, we should pay more attention to the false negatives for the next step classification.

**2. Node 32**

Node 32 is a very similar level node as node 36. It is also one of the topmost nodes in our decision tree. There are 15 leaves follow under this node. This node includes the keywords occurrence condition: NECK to be 0, LAC to be 0, FX is 0, CONT is 0, COLLISION to be 1, CHI to be 0, CONCUSSION to be 0, and HEAD is 0. In node 32, there are 48 data rows (87.27% of data) are classified to be cases, and 7 rows of data (12.73% of data) are classified as non-cases.

Node 32 uses “AND” to connect 8 conditions. As all the above conditions are satisfied at the same time, the data will be classified as 1, and otherwise, the data will be classified as 0 for my\_test. Node 32 is a high-level classification since it includes many conditions, and very few data could satisfy all of them concurrently. This classification does delivery some information. Since all the keywords occurrences will be 0 but COLLISION is 1, “COLLISION” is a very informative word to extract from the narrative. This node splits most of the data to be 1 for a head injury, and a few data to be 0. This means when all other keywords are missing but COLLISION is used in the description, it will very likely be a head injury. This split successfully increases the sensitivity, but we should recheck the miss-classification for the false positive rate. We could use “proc sql” with clause WHERE my\_def = 0 and my\_test = 1 to show the cases are wrongly classified as a head injury but actually not the head. We could avoid some of the keywords in the next split to improve the specificity if we could successfully reduce the false positive rate.

**3. Node 28**

Node 28 is a low-level node in our decision tree. There are 12 terminal-node leaves follow under node 28. This means that there has already been some successful classification based on the previous nodes. This node includes the keywords occurrence condition: R is 0, HD is 0, HEADACHE is 0, COLLISION is 0, CHI is 0, CONCUSSION is 0, and HEAD is 0. When all the conditions are satisfied, the prediction for my\_test will be classified as 1. Otherwise, the head injury prediction will be classified as 0. 12 pieces of data (92.31%) are classified into cases, and only 1 data row (7.69%) is classified as non-cases.

Node 28 uses “AND” to connect 7 conditions. As all the conditions are met, the head injury prediction test will be 1, and otherwise, the prediction test will be 0. This is still a long condition that very seldom narrative data will meet all the conditions simultaneously. Node 28 is only responsible for distinguishing 13 data. Since 92.41% percent of the data are classified as head injury, it is very possible the false positive rate to be high. Some of the non-head injury cases are tested to be cases. The specificity is affected by this node. We should check the data using “proc sql” with clause WHERE my\_def = 0 and my\_test = 1 to show the misclassification rows. The only false negative prediction should be correct. We need to pull out the false positive data to check the keywords extraction for the next level split to improve the specificity by reducing the false positive rate.

**4. Node 8**

Node 8 is a very low-level node in the decision tree model. There are only 2 leaves follow this node 8. This node includes keywords occurrence condition:

CONCUSSION is 0, STRAIN is 0, and HEAD is 0. If the three conditions are met at the same time, the prediction result will be a 0 (No head injury). Otherwise, the case will be classified as 1 (head injury occurred). Node 8 classifies 9 data rows (34.62%) to be cases and 17 rows of data (65.38%) to be non-cases.

Node 8 only uses “AND” to connect 3 conditions. Unlike the previous three nodes I talked about, this node classifies data more evenly than the previous nodes. Almost 35% of data are grouped as head injuries and 65% of data are classified as no head injury. If all three words, “CONCUSSION, STRAIN, and HEAD”, are missing, it will be unlikely to be a head injury. Both the false negatives and false positives could be produced in this node split since the conditions are too less and broad and it will be easier for data to satisfy this condition. The false-negative rate could be high here due to the easy-reach condition. We should post the data by using "proc sql" with clause WHERE my\_def = 1 and my\_test = 0. We should check the keyword extractions and add some conditions into the next level tree split to increase the sensitivity by reducing the false-negative rate.

**5. Node 7**

Node 7 is one of the last internal nodes in the decision tree split for head injury prediction. There is only one leaf following this node. 5 cases 100% of data under this node are classified as cases and no non-cases predicted. Node 7 has conditions:

CONCUSSION is 1, STRAIN is 1, and HEAD is 1. The prediction result is 1, which is the head injury predicted. Otherwise, the tree will classify as no head injury occurred.

The condition for Node 7 is almost the same as Node 8, but they may under different split branches. Node 8 says when three conditions are 0, the prediction result is 0, and node 7 says when three conditions are 1, the prediction result is 1. Node 7 is only responsible to classify 6 data, and all of them satisfy the three conditions. All three narratives include keywords of “CONCUSSION”, “STRAIN”, and “HEAD”. When all three keywords occurred in the narrative, it must be a head injury. Since node 7 classifies all the data into the true positives and is the last level internal node, and the conditions are the same as node 8’s conditions under different branches. I would presume the keywords are relatively accurate. However, we should still use "proc sql" to extract data by using WHERE my\_def = 0 and my\_test = 1 to check the false positives to see if any miss classification occurred. It will be unlikely to continue to improve the sensitivity or specificity, but this is a very good example node to visualize the split result for leaves of the decision tree model.

However, this node does help to correctly classify the True positives, which will be sensitivity. It will be effective for the dataset to predict the true positives since the condition is very strict to satisfy.