SL Project Checkpoint

Jaiden Callies (CS 4033) Alex Hunt (CS 5033)

1. Project Domain

For our supervised learning project, we are creating models to perform a classification task involving stroke diagnosis. We are attempting to accurately predict whether unseen patients are likely to have a stroke or not. The dataset we are using includes information on 5110 patients. The information recorded on each patient involves 11 different characteristics. These characteristics include a patient's gender, age, hypertension indicator, heart disease indicator, marriage status, work type, residence type, average glucose level, BMI, smoking status, and a stroke indicator.

2. Experiments & Results

2.1. Experiment 1

In our first experiment, we used the distance-weighted k-nearest neighbors learning algorithm to predict patients' potential stroke diagnosis. The dataset was split randomly into two sets. The training set included 75% of the original dataset while the testing set included 25% of the original dataset. For this experiment, we hypothesized that increasing the value of k when using the distance-weighted k-nearest neighbors learning algorithm would eventually result in decreased recall, specificity, precision, and balanced accuracy values. We used 13 different k values ranging from 1 to 25. The training set essentially included all possible neighbors for each testing set example. For each k value, we used the k-nearest neighbors learning algorithm to make our predictions. Additionally, we randomly split the data into the training set and testing set 10 different times. Each time we split the data, we ran our experiments using all k values and compiled the results. At the end of all experiments, we averaged the results over the 10 different runs.

2.2. Experiment 1 Results

As shown in Figure 1, it appears that our hypothesis is correct. Increasing the k value when implementing the k-nearest neighbors algorithm on our dataset did not result in higher performance metrics overall. In general, the distance-weighted k-nearest neighbors algorithm does not involve true learning. It is an instance-based learning algorithm in which the model passively accepts the data it is given. As such, we believe our results for this experiment are primarily due to the class imbalance of our dataset.

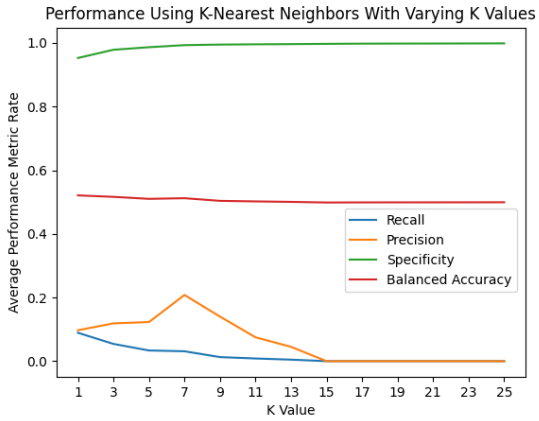

Figure 1. The average performance metric rate using the distance-weighted k-nearest neighbors learning algorithm for varying k values. The performance metrics used include recall, specificity, precision, and balanced accuracy. Each line represents different performance metric values over varying k values. The k value is simply the number of neighbors used in the algorithm.

There are significantly more patients in our dataset that do not have strokes than those that do. This explains the high specificity for all k-values since specificity is based on the number of true negatives and false positives. Using this algorithm, because the vast majority of examples include a negative stroke diagnosis, the number of true negatives is likely to be high and the number of false positives is likely to be low. Furthermore, other performance metrics seemed to decrease with increasing values of k. This makes sense. For any test example, increasing the number of neighbors used in the algorithm likely means that the majority of the example's neighbors will automatically be patients that exhibit a negative stroke diagnosis. Consequently, for this dataset, a model using a high value of k is more likely to produce a negative stroke diagnosis for all test examples. This would result in a higher rate of false negatives and a lower rate of true positives, thus explaining the decrease in recall and precision rates as k is increased. Ultimately, the recall and precision rates eventually reach a value of 0.0. At this point, it is likely that the model is only predicting a negative stroke diagnosis. Additionally, the balanced accuracy seems to be

skewed by the high rate of specificity. Ultimately, it appears that the optimal k value for this dataset lies somewhere between 1 and 9.

2.3. Experiment 2

For our second experiment, we focused on implementing and evaluating a decision tree model to predict the likelihood of stroke in patients based on various health indicators and lifestyle choices. The decision tree model utilized various configurations to understand how different parameters impact the model's accuracy and overall prediction capabilities. As seen in Figure 2, we experimented with different maximum depths of the tree—unbounded, 5, and 10—and two criteria for splitting nodes: Gini impurity and entropy.

2.4. Experiment 2 Results

As hypothesized, the unrestricted depth decision tree tended to overfit the training data, resulting in a high variance in its predictions on new, unseen data. This was evident from the fluctuating performance metrics across different depths. Shallow trees (depth of 5) generally offered a better balance between bias and variance, optimizing the model's ability to generalize while maintaining a decent level of accuracy. The Gini impurity and entropy criteria provided slightly different results, with entropy tending to slightly outperform Gini in deeper trees, suggesting it might be handling the dataset's inherent complexities more effectively.

These experiments revealed several avenues for further exploration. It became clear that managing the depth of the tree is crucial for preventing overfitting and improving model robustness. Future experiments could explore more granular depths between 5 and 10 to pinpoint the optimal depth for this dataset. Additionally, integrating ensemble methods such as random forests or boosting could be investigated to assess their potential in enhancing prediction accuracy and stability by aggregating the decisions of multiple trees.

3. Challenges

Jaiden is primarily having issues evaluating model performance effectively. By nature of our dataset, it is difficult to use traditional methods of evaluation, such as empirical risk. As a general rule, the majority of individuals are unlikely to have certain medical anomalies, such as a stroke. Therefore, the dataset itself has a class imbalance. Normally, low empirical risk would correspond with a better model. However, in this case, due to the class imbalance, low empirical risk may not provide us with any information about the efficacy of our model. The model could simply have low risk because it only predicts the majority class. At the moment, Jaiden is using precision, recall, specificity, and balanced accuracy to evaluate performance. In subsequent

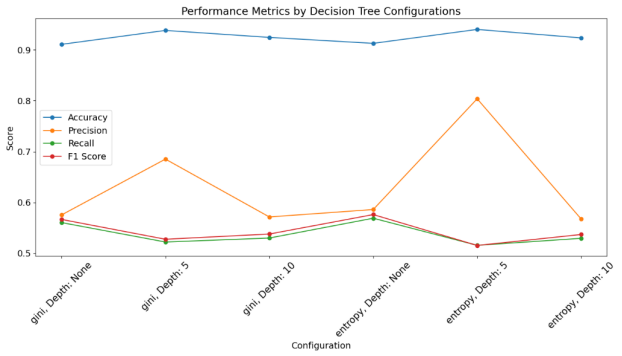

Figure 2. This graph displays the accuracy, precision, recall, and F1 score for various configurations of a decision tree model. Each line represents one of the four metrics, evaluated across different decision tree setups defined by node splitting criteria (Gini impurity and entropy) and tree depth (unbounded, 5, and 10). The "unbounded" configuration allows the tree to grow until all leaves are pure or until all leaves contain less than the minimum split samples. The configurations with depths of 5 and 10 introduce constraints to curb overfitting, with deeper trees typically capturing more complex patterns but also risking overfitting compared to shallower ones. These settings are chosen to explore the balance between model complexity and generalization ability in predicting stroke likelihood. The metrics are calculated on a macro scale to give equal weight to each class, addressing class imbalance. This visual representation aids in understanding how different tree depths and splitting criteria impact the overall effectiveness of the model in a medical prediction context.

experiments, she may explore other performance metrics.

During Alex's experiments with decision trees to predict stroke likelihood, one significant challenge he encountered was the difficulty in finding reliable and detailed resources online that could guide the nuances of advanced decision tree configurations and optimizations specifically suited to medical datasets. This obstacle made it harder to refine the models beyond basic implementations and assess the impact of more sophisticated techniques that could potentially enhance model accuracy and robustness.

4. Next Steps

The next step for our project is for each of us to implement one more supervised learning method. Jaiden needs to implement naive Bayes learning, and Alex needs to implement random forests. Besides this, our main priority is to find three approaches to compare our performance to. Additionally, we still need to select more papers related to our experiments to discuss in the final report. Alex needs to find three papers, and Jaiden still needs to find one more. Lastly, once we finish the other steps, we must combine all of our work into a final report.