Lin Ouyang

CS 5350/6350 Machine Learning,

Fall 2024 – Projects

Mid-term Report

**What I did?**

**Data Processing and Feature Engineering**

A critical aspect of my approach was effective data preprocessing, especially the transformation of numerical features and the handling of missing values. I chose to convert numerical features into binary values based on the median value within the training set, creating a threshold that divides each feature into two categories: values above the median were set to 1, while those below were set to 0. This binary transformation simplified the feature space, making it easier for decision tree algorithms to evaluate splits, while potentially enhancing model robustness by reducing sensitivity to minor fluctuations in the data.

The treatment of missing values represented by "?" evolved as I progressed. In my initial submissions, I treated the "?" symbol as a unique value, meaning it became part of the attribute's possible values during training and testing. However, after observing the model’s performance and analyzing the data, I revised this strategy in later submissions by replacing missing values with the mode (most frequent value) of that attribute in the training set. This approach ensured greater data consistency and reduced the likelihood of the model treating “?” as a distinct category, which could lead to spurious patterns and suboptimal splits in decision trees.

**Model Selection and Iterative Optimization**

I explored decision tree models and random forest models. For the decision tree model, I used binary prediction, that is, directly predicting the TARGET variable as 0 or 1 for each test set ID. For the random forest model, I explored the probability prediction method. For each test set ID, output the probability value of its TARGET variable being 1, ranging from 0 to 1:

**First and Second Submissions - Decision Tree Model**

Method: In my initial attempts, I used a basic decision tree model with a maximum depth of 14 to control for potential overfitting. The first submission employed the Gini index as the criterion for splits, while the second used entropy, providing a different metric to assess purity in each node.

Results: The initial submission, using the Gini index, scored 0.71983 on the Kaggle leaderboard. The second submission, using entropy, yielded a slightly improved score of 0.72315, indicating that entropy was more effective in this context.

**Third Submission - Decision Tree Model**

Method: In this submission, I further optimized the decision tree by refining the approach to handling missing values. Instead of treating “?” as a standalone attribute value, I used the mode for each attribute to fill in missing entries. This slight modification enhanced data uniformity and allowed for a more natural division of instances across nodes, which I hypothesized would aid in generating more meaningful splits.

Results: This adjustment brought a marginal improvement to 0.72335, reinforcing the value of carefully considered data imputation methods in achieving optimal model performance.

**Fourth and Fifth Submissions - Random Forest Model**

Method: Moving to a more robust ensemble approach, I adopted a random forest model for the fourth and fifth submissions. Without setting a maximum depth, I trained 500 trees for each forest to capture complex patterns in the data. To enhance generalizability, I applied bootstrapping, drawing a subset of samples with replacement. The fourth submission involved 10,000 training samples, while the fifth used 20,000 samples, allowing me to test the effect of sample size on accuracy and model stability.

Results: The fourth submission demonstrated a significant performance increase, scoring 0.86783 on the leaderboard. This leap underscored the power of ensemble methods like random forests in capturing nuanced patterns within the data. However, the fifth submission, using a larger sample of 20,000, resulted in a slight drop to 0.85245, indicating that increased sample size can introduce additional noise or redundancy, potentially reducing model efficiency. This experience highlighted the need for balance in sample selection to ensure that the model remains both accurate and resilient to overfitting.

**Sixth to Eighth Submissions**

In the sixth to eighth submissions, I continued to refine the random forest model by adjusting the training sample size based on the fifth submission to further examine the impact of sample size on model performance. In the sixth submission, I used 15,000 training samples, achieving a score of 0.85806. For the seventh submission, I reduced the sample size to 5,000, which improved the score to 0.87771. In the eighth submission, I further reduced the sample size to 1,000, resulting in a final score of 0.88871. This outcome suggests that, for the current dataset, reducing the sample size can enhance the model’s generalization ability and prevent the interference of redundant data. These experiments further demonstrate the critical balance between sample selection and model performance in achieving optimal results.

**Plans for the rest of this semester**

**Improve The Random Forest Method**

During the remainder of the semester, I plan to improve the random forest model to further improve prediction performance. Through the experimental results of the first eight submissions, I observed a significant impact of model selection on the final score, especially the transition from a single decision tree to a random forest, which resulted in a significant improvement in model performance. Based on this improvement, I realized the potential of random forests in handling complex data sets and wanted to continue to optimize their performance.

Specifically, I will explore different hyperparameter combinations, including the number of trees, maximum depth, and sample sampling strategy to find the best configuration to improve the generalization ability and accuracy of the model. By analyzing the impact of different parameter configurations on model performance, I can gradually iterate and move towards higher scores.

**Use Other Machine Learning Methods**

In addition, I also plan to introduce other machine learning methods in the prediction task to further explore the effects of different models. For example, the least mean square regression learned in class. At the same time, I will also draw on deep learning methods such as convolutional neural networks learned during undergraduate studies. By integrating what I learned in class and undergraduate knowledge, I hope to find the best combination among multiple models to further improve the accuracy of prediction and achieve better results in this competition.