Group e1 Final Project Report

Colab Link:

https://colab.research.google.com/drive/15A6lsvobK1744yssiJkGmrutPTYWe8S6?usp=sharing

Introduction

The foraging and consuming of wild mushrooms is an activity that humans have participated in throughout history, and though in many cases it is no longer required for survival, it remains a popular hobby. This hobby contains a significant element of danger, with the challenge of distinguishing a poisonous from an edible mushroom being quite subtle and complex. One must be able to learn complex patterns between the physical features of the mushroom to determine if it is edible, and just a single mistake could lead to serious illness or even death.

Learning complex patterns within a dataset is a task where machine learning shines. With modern computational power, machine learning algorithms can detect patterns in datasets that humans, even experts in the given field, cannot easily find. In this case, we have a binary classification task (determine whether a mushroom is poisonous or edible) that depends upon 22 categorical features (the physical characteristics of the mushroom). This is quite a daunting task for a human, especially amateur mushroom hunters who have limited knowledge and experience identifying mushrooms, but an easily manageable task for a suitable machine learning model.

Methods

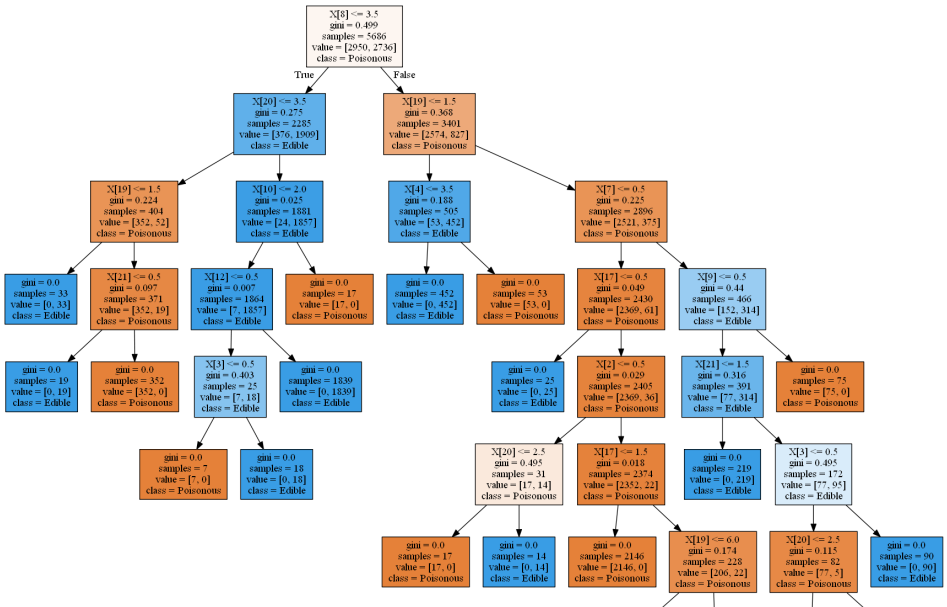
The dataset we used is the UCI Machine Learning Repository Mushroom Data Set, which is a compilation of data from the Audubon Society Field Guide to North American Mushrooms. This dataset is available as a csv file on Kaggle.com, and this is what we directly used to

develop, train, and test our models. It contains 8124 instances, each of which is a unique mushroom with 22 physical features, ranging from cap shape to stalk color, and a label of poisonous or edible.

On Kaggle, we found a variety of previous approaches to the task of determining if a mushroom was poisonous or edible given its physical features. Using a Random Forest Classifier, one user was able to obtain a 100% accuracy score on a test set which contained 33% of the original dataset. A different user obtained 99.7% accuracy on a test set 25% the size of the original data using KNN classification, and another achieved 100% accuracy on a test set of 20% using an XGBoost classifier.

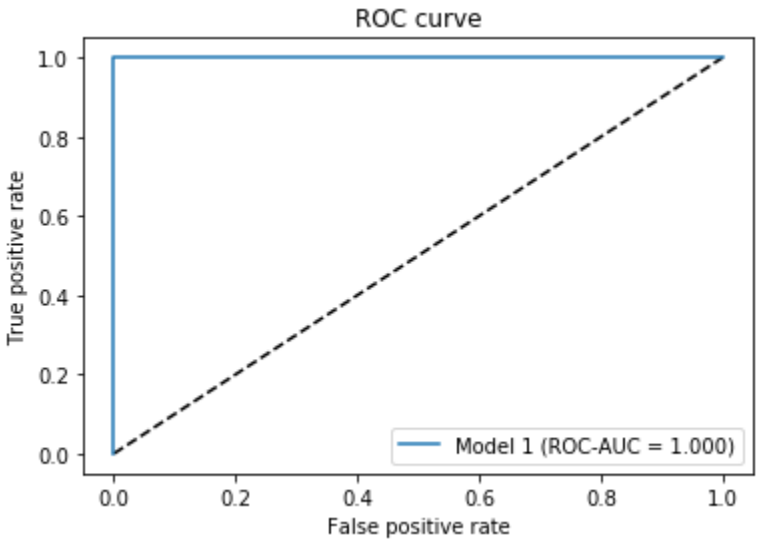
Clearly, we soon realized that we couldn't simply build a model with the goal of better performance than other models in the literature, as you can't get much better than 100% accuracy! Instead we asked ourselves, "can you obtain this level of performance in a new way, especially a way that requires less computational power?" The answer to this question was ultimately found within decision trees.

Initial testing on a default DecisionTreeClassifier showed promising results, but not at the level of the current literature. To optimize the model, we used GridSearchCV, running a total of 20580 fits to find the best combination of max_leaf_nodes, min_samples_split, and max_depth that fit the training data. The final decision tree is shown in the following graph:



Results

We are now ready to answer the key question: “how does the model perform?” We were able to obtain a 100% accuracy score on a test size which was 30% of the original dataset, outperforming some of the even more complex models found in the literature. The ROC curve appears as a triangle with an AUC value of 1, as expected:

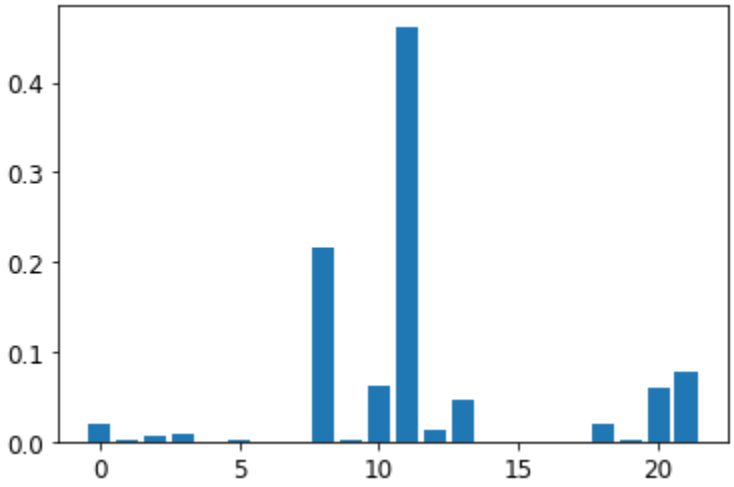


While we were satisfied with the performance of this model and succeeded in demonstrating our original interest that an even simpler model could achieve a high accuracy on this dataset, we saw that some more could be done. We thought about how we could look at this dataset in new ways, how we could extract more information and usefulness out of it, and this brought us to analyzing feature importance/interdepence

Side Projects

Feature importance/interdependence

We checked the interdependence of the features by exchanging the role of the label (poisonous/edible) with each feature, using the decision tree model on 22 additional, usually multi-class, classification problems; this allowed us to see how important the features, including the original label, were for classifying mushrooms by, for instance, habitat or odor. One notable conclusion was that stalk shape was especially important for predicting population.



Feature importance for predicting population. Tallest bar corresponds to stalk shape.

This was an interesting exercise in seeing how a dataset can be extrapolated and used to gain more information about the things that the data describes, in this case being mushrooms. It's a showcase not only of how machine learning can help us pick up on subtle patterns within data, but how you can use a dataset in more ways than just the one you originally had in mind, which

in our case was edibility classification. Reusing a dataset to approach and explore multiple ideas can significantly save on the time, money, and effort required to make new, functional datasets.

Image Classification

We had a bit of extra time on our hands and decided it could be good to explore the task of mushroom image classification. Specifically, we wanted to build a model that could accurately predict the genus of a mushroom in an image. The inspiration behind this was the idea of a smartphone app that could determine if a mushroom is edible or poisonous: if you had an app that used a machine learning model to identify the type of mushroom in an image and some of its features, you could feed that information into the decision tree model we showcased earlier to get an estimate of its edibility. Such an app could become a mainstream, highly beneficial addition to the mushroom hunting community that would make the hobby much safer and accessible.

Another Kaggle dataset was used to explore this task, as it contained 6714 mushroom images of 9 different genuses. Though we were able to find different ways to load and augment this data, we were unfortunately never able to construct a model with any significant amount of performance. Nonetheless, we found in searching through the literature that others were able to obtain accuracies of around 80% using ResNet transfer learning.

Given a more optimized dataset and a better tuned model, there appears to be a real possibility that such an app could become a reality in the not-so-far future. Though we were unable to build a high quality model at this time, it was exciting to explore and think about how such a powerful and influential piece of technology could be soon developed using machine learning and the techniques we've learned.

(At the end of our notebook you can see some of the rough code of what we briefly tried, but it is certainly not the focus of this project/report, especially considering that the code cannot be run properly in colab. This is due to difficulty in making the files accessible in colab).

Conclusions

In this report, we've discussed how a decision tree classification model was able to obtain 100% accuracy on the complex task of mushroom classification. This showcases the powerful performance that decision trees can yield given a suitable task, and this serves to remind us of the usefulness that such algorithms have. Additionally, our explorations of feature interdependence serve as a brief demonstration of how a dataset can be used multiple ways outside of the primary task, yielding greater knowledge without the need for any new data.

Datasets used:

https://www.kaggle.com/uciml/mushroom-classification (edibility classification, feature interdependence)

https://www.kaggle.com/maysee/mushrooms-classification-common-genuss-images (image classification)