**MUSHROOM CLASSIFICATION**

MA 440: Final Project

Dr. Hong Liu

Cade Roberts, Lance Cross, Naomi Sterlingsdottir, Ying Zheng

# Abstract

An analysis was performed on the UCI and Phillips University mushroom dataset to classify edible and inedible mushrooms. The data from the two sources were combined together, made tidy and then augmented with dummy variables to ensure all metrics were accounted for using integer inputs. The classifications are based on physical features of the mushrooms such as its cap, gills, stalk, rings, and etc. Each classification model developed was measured by its accuracy, confusion matrix, recall, precision, f-score, kappa score, and AUC. After training, KNN, SVM, Adaboost, decision tree, and random forest, they were all above 90% accuracy, but the random forest classifier stood out as the best with an astounding 99% classifying accuracy in our testing, with a 25% training percentage.

*Keywords:* KNN, SVM, Adaboost, Random Forest, Decision Tree

# Background

There are over 14,000 identified species of mushrooms, many of which are widely enjoyed as a culinary delicacy, and others that even a gram of which could result in a slow and painful death. Naturally, this rather stark dichotomy necessitates a level of confidence in your decisions should you choose to go out and forage for your own mushrooms, whether it be for culinary or medicinal purposes. Types of mushrooms vary depending on regional location, and because of this, specific species identification can be difficult, and the novice mushroomer should stick to genus’ of which every species is edible, such as chanterelles, and most morels. Unlike these quite friendly genuses, there are much more dangerous ones that include some edible, and some deadly mushrooms, one such genus being Amanita, which contains the very tasty Amanita Fulva, to the very deadly Amanita Virosa. The combined data set we analyzed contains many species from the amanita genus as well as Lepiota, both of which contain edible and inedible mushrooms. By analyzing specific aspects of the mushroom's appearance, cap color, gill attachment, gill color, and others, a model could be trained to distinguish poisonous mushrooms from edible ones.

# Problem

The goal of this project is to develop multiple classification models and find the best performing model that can classify whether a mushroom is edible or not. The data was gathered by combining two datasets and then preprocessed for training. The models developed were binary classification models such as KNN and Decision Tree. Each model was tuned to find the most optimal versions of each model. Tuning methods such as GridSearchCV, n\_estimators, and loops were used to tune these models. To validate the models and compare them, we used metrics such as accuracy and Cohen’s kappa.

# Dataset

The mushroom dataset used was obtained by the UCI Machine Learning Repository and Phillips University Marburg. This dataset contains two mushroom families that include the Agaricus and Lepiota containing 58 attributes. These attributes are the characteristics of the mushrooms which allows the variety of attributes to be used to detect whether mushrooms are edible or inedible.

Most machine learning algorithms do not process categorical data therefore the dataset was transformed into numerical form where a string is assigned a numerical value in place of the string. Since the observations are ordinal, dummy variables can be used to represent categorical data by assigning 0’s and 1’s to the data. This helps the computer understand the relationship between the observations and attributes based on the specific algorithm. The number of unique observations per attribute determines the number of columns in the dummy variable dataset.

# Model Development

This project used a variety of machine learning algorithms, including two ensemble methods such as Decision Tree, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression, Random Forest and AdaBoost. Each model will yield different accuracies and results based on dataset due to each model having a unique way of classific. Ensemble method goals are to improve classification accuracies by aggregating predictions from multiple classifiers hence they should outperform other machine learning algorithms. All these models require numerical data hence converting the tidy table into a dummy variable dataset.

Decision trees are used for regression and classification where decisions are based on conditions of any features in the dataset. Each decision tree begins with a root node and entropy/Information gain are used as a basis to determine a specific set of conditions that determine when a new nodes or leaves are created. Advantages using this model is it can provide insight into why it made the predictions it made based on entropy/information gain and it makes no assumptions on the distribution of the data. However, overfitting this model can yield higher accuracy but then creates a complex tree which is where pruning can be used. This algorithm is not recommended on complex datasets otherwise it will begin to lose valuable information such as this dataset.

K-Nearest Neighbors (KNN) is also a supervised, lazy machine learning algorithm that uses distance or proximity as a measure of classifying data due to the assumption that identical things exist near each other. Advantages with this algorithm is it does not require training to make predictions and works based on relatively small datasets. The K value and distance function, which are hyperparameters, can be tuned to increase accuracy. However, this algorithm is susceptible to noise and outliers causing predictions to be inaccurate. This algorithm requires high memory and computational power compared to most other machine learning algorithms making it a less desirable algorithm to use. In this case we have no outliers in out dataset making this algorithm suitable to predict edible or inedible mushrooms.

Support Vector Machine (SVM) is also a supervised machine learning algorithm that creates a hyperplane (3D) or line (2D) to separate data into classes. Some advantages it has is it is less susceptible to overfitting since it handles outliers and uses a kernel trick to solve non-linear problems. Note that SVM can outperform its cousin algorithm, KNN, if there is a larger number of features and smaller training dataset. A few disadvantages are it cannot handle massive datasets, however, in this case the data set size was suitable for this algorithm to process the data with high accuracy. This algorithm does not provide a clear explanation for the probability of specific points classified in certain locations around the hyperplane.

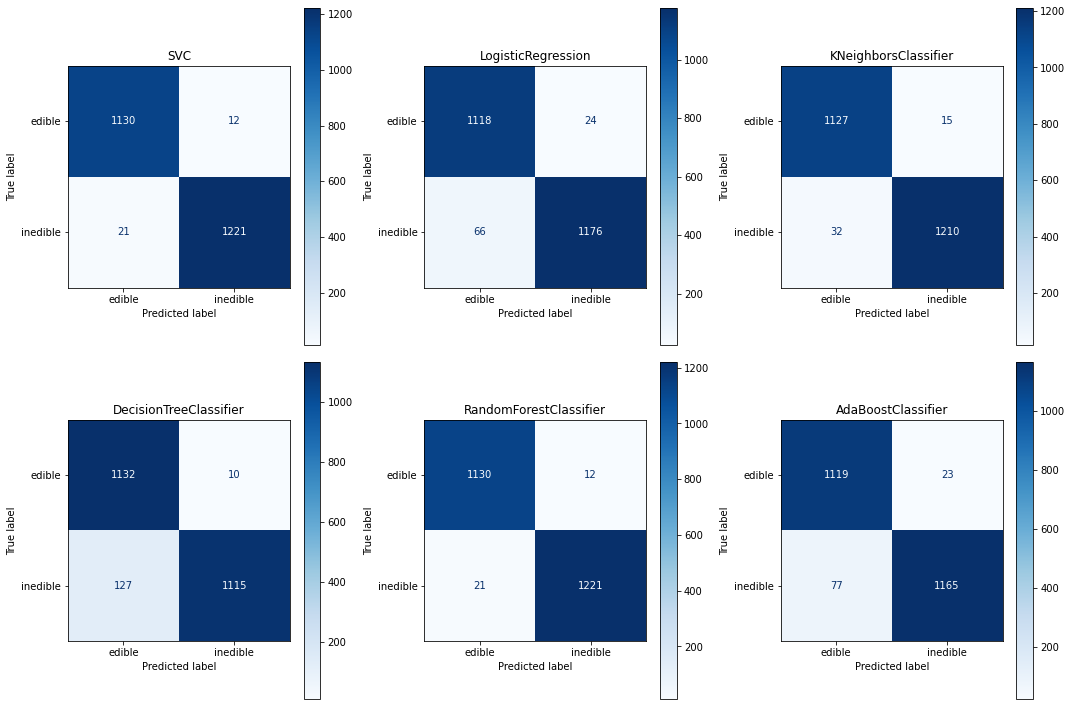
Logistic Regression is commonly mistaken as a regression model, however, it is a classification model where it uses a logistic function to create a binary output model. A sigmoid function is used to plot where data points will be classified as based on their probability and binary output. With two hyperparameters, the model can be tuned to achieve high accuracy. This algorithm has the advantage of being able to execute faster than other algorithms and provides explanation of independent variables with dependent variables. The major disadvantage this algorithm has is it cannot be applied to non-linear classification problems and outliers drastically affect the accuracy of the model. This algorithm works well for multiclass classification for datasets such as ones used in this project.

Random Forest is an ensemble method in machine learning that builds multiple decision trees and uses a Majority Voting for classification and regression. This algorithm greatly improves accuracy due to solving overfitting with its majority voting system. With datasets such as this mushroom dataset, it is able to randomly select attributes and build trees (hence the name random forest) independently to find the best possible tree whereas a decision tree only creates a single tree and relies on hyperparameters to tune for accuracy. It is expected than ensemble methods to outperform classic machine learning models due to averaging/majority voting system and a dataset such as this mushroom dataset can provide the most accurate tree to accurately determine whether mushrooms are edible or inedible.

Lastly, Adaptive Boosting (AdaBoost) is a boosting algorithm that combines weak classifiers into a single strong classifier. This algorithm is used to boost the performance of any machine learning algorithm, specifically Decision Tree and Random Forest, which is one of its greatest advantages. AdaBoost is an ensemble of decision stumps based on attributes similar to Random Forest except it can be called Forest of stumps which are not great at making accurate decisions on their own. With a voting system, bigger stumps get a final say in the final classification. Since stumps take into account the mistakes previous stumps made, it provides a more accurate classification of the attributes for determining which features are the most dominant in edible or inedible mushrooms.

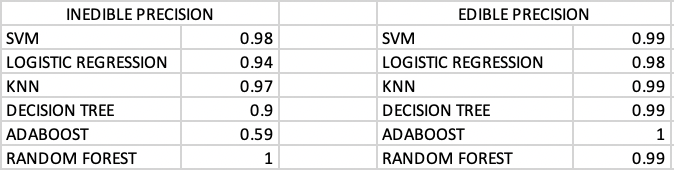
# Metrics

In order to compare the model the following metrics were performed on the models: confusion matrix, recall, precision, accuracy, f-score, kappa score, and AUC. A confusion matrix was printed for each model. The matrix provides the values for True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP) respectively. These calculations measure the performance of the model by specifying how many correct and incorrect yes or no predictions the model had (Brownlee, 2020). Typically, the higher the TN and TP are the better the model performed. **Figure 1** provides the confusion matrices for each model. The SVM and Random Forest model performed the best when comparing the confusion matrices.

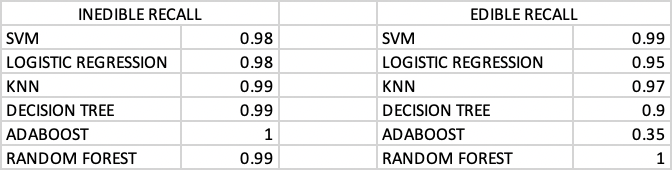


**Figure 1**

Precision and recall are both derived from the values obtained from the confusion matrix. Precision is the ability of a classification model to identify only the relevant data points. Recall is the ability to find all relevant instances in a dataset. Since the dataset is an imbalanced dataset, a sklearn classification report was run for each model which provided the precision and recall for each label. In general, the precision and recall for each model are similar in values except for Adaboost with the lowest values. For precision Adaboost predicted the edible data better than the inedible data and for recall Adaboost predicted the inedible data better than the edible data. Overall, Random Forest had the best precision and recall scores.

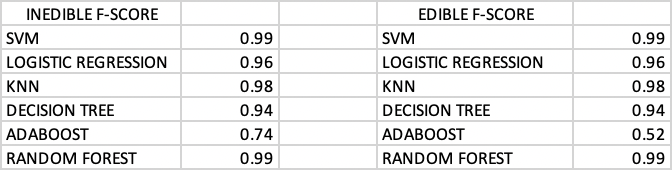


**Figure 3**



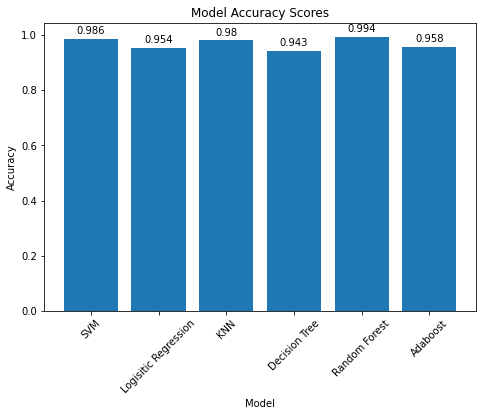
**Figure 4**

F-score is derived from the precision and recall scores. It’s the harmonic mean of precision and recall. The higher the score, the better the classifier. Since Random Forest performed the best and Adaboost performed the worst, the F-score reflects that performance with Adaboost having the lowest scores and Random Forest and SVM with the best scores.



**Figure 5**

Accuracy is the number of correct predictions divided by the total predictions. **Figure 6** shows all the accuracy scores for each model. According to the figure, the Random Forest model has the highest accuracy compared to the other models.



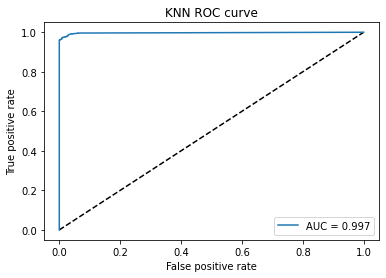
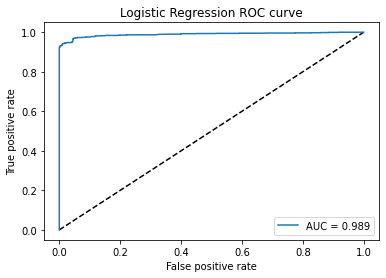
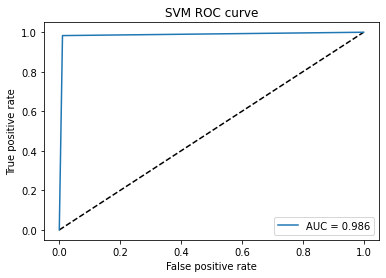
**Figure 6**

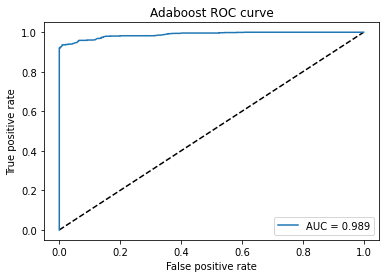
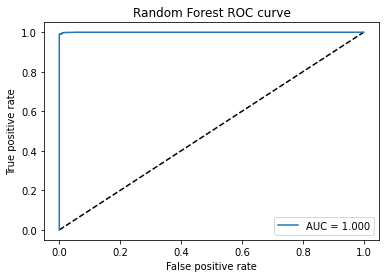
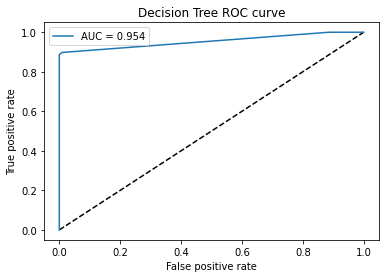
As mentioned previously, the mushroom dataset is imbalanced. When a dataset is imbalanced accuracy is not enough to determine if the model is performing well. Cohen’s kappa is a measure of interreliablity between two raters. It takes into consideration imbalanced datasets by ignoring predictions that are possibly determined by random (Widmann, 2021). The model with the highest Kappa score is again Random Forest.

# 

**Figure 7**

Finally, the last metric that was used on each model was ROC curves and its AUC scores. ROC curves are plotted using precision and recall; it is a visualization of precision and recall. AUC sums up the ROC curve in one number and generally helps determine how well the model is distinguishing between the classes (Bhandari , 2020). Random Forest proved to have the best AUC with a perfect 1.00



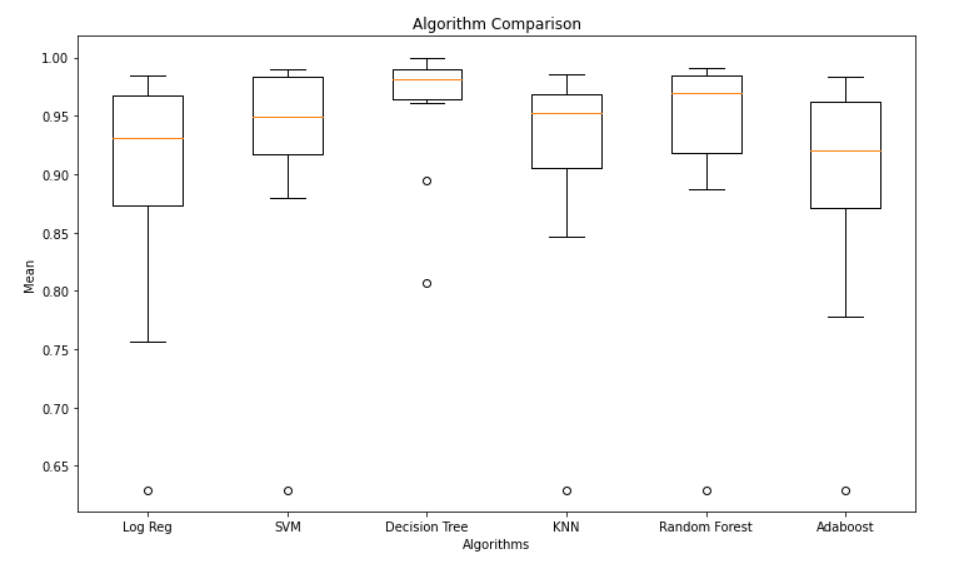


# Results

Overall the model with the best results is Random Forest. It had the highest values for each metric that was run on it compared to the rest of the models with a kappa score of .987 and an accuracy score of .994. Although Random Forest was the best model, the other models that were developed had close values to Random Forest with SVM coming close to the results that Random Forest had. The worst model was Adaboost which had lower values than the rest with a kappa score of .916 and accuracy score of .958. It is expected for Random Forest to yield the highest accuracy due to it being an ensemble method where it uses a major-vote to determine the best decision tree based on accuracy.

# Discussion

When comparing models, using a boxplot required a cross fold validation in order to accurately compare the models to one another. The cross fold validation method is where the average of the accuracies of a model is taken over a certain amount of iterations to gauge how accurate the model actually is. After running all 6 models through the 10-cross fold validation, the accuracy values were lower compared to the fine tuned models. Random forest had the highest accuracy compared to all other models whereas Decision tree performed the best when using the 10-fold cross validation method. A possible reason for Random Forest to not perform well is that it is an ensemble method that uses a majority-vote to determine the best and most accurate decision tree. Because of that Random Forest will have a lower accuracy due to it taking an average of all decision trees, including trees that are inaccurate.



Due to this dataset containing only two species of mushrooms, adding more contrasting species would allow more classifications of mushrooms and add additional training data. With only 2 species, it is easier for the models to be fine tuned for accuracy. The more data a model, the more precise and accurate the model can become based on the number of attributes, therefore, working with other attributes such as

As seen within the correlation matrix, there were four main attributes that contributed to deducing whether mushrooms were edible or inedible; gill color, veil color, gill attachment and veil type. It is believed these attributes contributed to the solution of edibility is that this dataset mainly contained gilled mushrooms which validates why the four previously mentioned attributes have the strongest correlations. Attributes that would be used in the future work would be cap shape and habit since mushrooms come in all unique shapes and locations which would better determine if they are or aren’t edible.

Possible further improvements that could be made would be to include photos of certain mushroom species as entries and utilize Neural Networks to improve the classification with more qualitative data. Using deep learning would increase the reliability of model in a real world application which is how other AI apps are used for when classifying flowers or types of plants.

# Conclusion

Our model did a fantastic job at accurately predicting which mushroom would be poisonous and which would not. At a 99% accuracy rate our model error could not even be considered statistically significant and could be used with expert judgement as a valuable tool in identifying edible mushrooms. Utilizing Neural networks moving forward as mentioned in our discussion would be ideal as it could possibly be turned into an app that could identify mushrooms in the wild with a phone camera. In this project we were able to combine 2 data sets from different time periods into a tidy dataframe, and pick out the metrics that most influenced their identification. After the final data set was codified, we then trained 6 different models to find which one would be the most accurate to the testing data. Each of the ROC curves and confusion matrix’s were compared, and the Random Forest model ended up being the most accurate, but nearly every model was above 90% accuracy.

# 

# References

Bhandari , A. (2020, July 20). *AUC-Roc Curve in machine learning clearly explained*. Analytics Vidhya. Retrieved December 9, 2021, from <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>.

Brownlee, J. (2016, June 1). *How To Compare Machine Learning Algorithms in Python with scikit-lean*. Machine Learning Mastery. Retrieved December 2, 2021, from https://machinelearningmastery.com/compare-machine-learning-algorithms-python-scikit-learn/

Brownlee, J. (2020, August 14). *What is a confusion matrix in machine learning*. Machine Learning Mastery. Retrieved December 7, 2021, from <https://machinelearningmastery.com/confusion-matrix-machine-learning/>.

Desarda, A. (2019, January 17). *Understanding AdaBoost*. Towards Data Science, Retrieved November 20, 2021, from <https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe>

Thanigai, K. (2020, May 25). *Adaptive Boosting or AdaBoost Algorithm*. Medium. Retrieved November 30, 2021, from https://medium.com/@ktv0303/adaptive-boosting-or-adaboost-algorithm-4e8b9b372556

Widmann, M. (2021, October 31). *Cohen's kappa: What it is, when to use it, and how to avoid its pitfalls*. The New Stack. Retrieved December 9, 2021, from https://thenewstack.io/cohens-kappa-what-it-is-when-to-use-it-and-how-to-avoid-its-pitfalls/.

Varghese, D. (2018, December 18). *Comparative Study on Classic Machine Learning Algorithms*. Towards Data Science. Retrieved December 10, from https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222