CSE514A Project 2 Final Report

Autumn Xu, Chris Gu, Zhaohua Zheng

Introduction

This project uses the UCI Mushroom dataset to explore two classification problems: predicting 1. the poisonousness (i.e., whether it is poisonous or edible) and 2. the odor of a mushroom based on remaining observable features. The raw dataset consists of 8124 instances and 23 categorical attributes, each representing different characteristics of mushrooms such as cap shape, color, gill attachment, and habitat. All features are nominal and require preprocessing/encoding for effective use in machine learning models.

The first classification problem aims to predict the poisonousness of a mushroom ("Poisonousness"), which is formulated as a binary classification task that assigns each mushroom to one of two classes: poisonous (p, positive class) or edible (e, negative class). This task carries practical significance and high stakes, as misclassifying a poisonous mushroom as edible could result in severe health consequences.

The second classification task involves predicting the odor of a mushroom, a categorical variable with nine distinct classes: almond (a), anise (l), creosote (c), fishy (y), foul (f), musty (m), none (n), pungent (p), and spicy (s). Accurately predicting odor has practical value in situations where sensory information is unavailable or difficult to assess, such as when working with dried samples, images, or automated systems, or when someone infected with upper respiratory disease cannot smell the mushroom in the wild. A model that can infer odor from physical characteristics can support educational tools, scientific cataloging, and assistive technology for mushroom identification.

For both problems, we apply multiple techniques (Decision Tree, Random Forests, CatBoost, Artificial Neural Networks (ANN), K-nearest neighbors (KNN), and Naive Bayes Classifier) upon 5-fold cross-validation to model relationships between the mushroom's physical traits and its poisonousness or odor. Given the number of features and thus high dimensionality introduced by categorical feature encoding, we employ Principal Component Analysis (PCA) to reduce the dimensionality of our data and compare model performances. This step aims to improve computational efficiency, mitigate overfitting, and enhance the generalization ability of the models on unseen data.

Results

The results of our experiments indicate that the two classification problems—predicting mushroom poisonousness and predicting mushroom odor—differ significantly in predictability and complexity. Overall, predicting poisonousness proved to be much more straightforward and yielded higher accuracy across all six tested models, regardless of whether the original dataset or the PCA-reduced dataset was used. Most models achieved testing and training accuracies above 0.99 in the poisonousness prediction task, with the only slight dip being the Random Forest model before dimensionality reduction, which still maintained a high testing accuracy of 0.9628. In contrast, odor prediction was substantially more difficult, with training and testing accuracies generally falling in the range of 0.90 to 0.96. This discrepancy was due to the fundamental

difference in task complexity—poisonousness classification is binary, while odor prediction is a multiclass prediction problem involving nine categories. The poisonous task likely benefitted from a clearer decision boundary between “edible” and “poisonous” mushrooms, whereas odor classification involves more nuanced differences, leading to overlapping feature distributions and lower classifier separability.

Not all models performed equally well across the two tasks. Models like K-Nearest Neighbors and Naive Bayes Classifier consistently achieved the shortest training times, typically under 0.01 seconds, which is attributable to their minimal training overhead. KNN had no training phase beyond storing the data, and NB relies on basic statistical estimations. However, these models also showed performance limitations, particularly in the more complex odor classification task. For instance, Naive Bayes experienced a noticeable accuracy drop of over 0.01 in both training and testing when dimensionality reduction was applied, suggesting a vulnerability to information loss during PCA. In contrast, more complex models such as Random Forest, CatBoost, and Artificial Neural Networks (ANN) generally demonstrated slightly stronger performance on the odor task than KNN and Naive Bayes Classifier, but required longer training time due to iterative learning processes or ensemble approaches.

Hyperparameter tuning results also revealed that models responded differently to the complexity of each classification problem. For example, for Decision Tree, the optimal cost-complexity pruning parameter was lower for poisonousness, allowing the tree to grow more before being pruned, while a higher pruning parameter was better suited for the more complex odor task, promoting generalization. Similarly, Random Forest required a lower mtry value for poisonousness, implying that only a few features were critical for prediction. In contrast, odor prediction benefited from higher mtry values, suggesting that a broader set of features was needed to make accurate distinctions among the nine odor classes. KNN also exhibited this behavior, with optimal k values being significantly smaller for poisonousness and much larger for odor, again underscoring the greater complexity of the multiclass problem.

The impact of dimensionality reduction using principal component analysis varied across models and tasks. In terms of performance efficiency (indexed by runtime), for both problems, PCA produced shorter or comparable training and testing runtime for all models except for Decision Tree, where runtime increased. The pattern of accuracy was more inconsistent across models. Specifically, it achieved comparable performances for both problems in Decision Tree, Random Forest, and KNN; in contrast, it degraded the performances in CatBoost and Naive Bayes Classifier. CatBoost performed worse with PCA possibly because the transformation into continuous principal components had stripped away the category-specific logic that underpinned its effectiveness. Similarly, Naive Bayes Classifier saw reduced accuracy possibly because the simplified feature distributions created by PCA accounted for the interdependence of features and thus interfere with the assumption of conditional independence made by the model. Finally, the effect of PCA on accuracy was task-dependent for ANN, where it improved the performance for the simpler poisonousness prediction task by reducing feature redundancy, but degraded the performance slightly for the odor classification task, likely because PCA removed subtle input variations needed for distinguishing between the more complex odor labels. Concluding from both the perspectives of efficiency and accuracy, the benefit of dimensionality reduction appears to be model-specific and context-dependent rather than universally advantageous.

We made observations and derived recommendations by analyzing the contributions of different features to the prediction of the response variables. This analysis provided insight into which attributes were most influential for each classification task, enabling more informed model interpretation and potential feature selection strategies.

First, for the poisonousness classification task, we observed that odor was the most influential feature. Specifically, mushrooms with a foul odor (odorf) were strongly associated with being poisonous, while those with no detectable odor (odorn) were typically edible (Figure 1, Random Forest output). This finding aligns with real-world intuition and reinforces odor as a highly discriminative attribute. In addition, certain tactile and visual characteristics also showed strong associations with edibility. Features such as a silky stalk surface below the ring, a tapering stalk shape, and a silky stalk surface above the ring were all indicative of poisonous mushrooms. These results suggest that a combination of sensory traits—both smell and physical appearance—can serve as reliable indicators of mushroom toxicity.

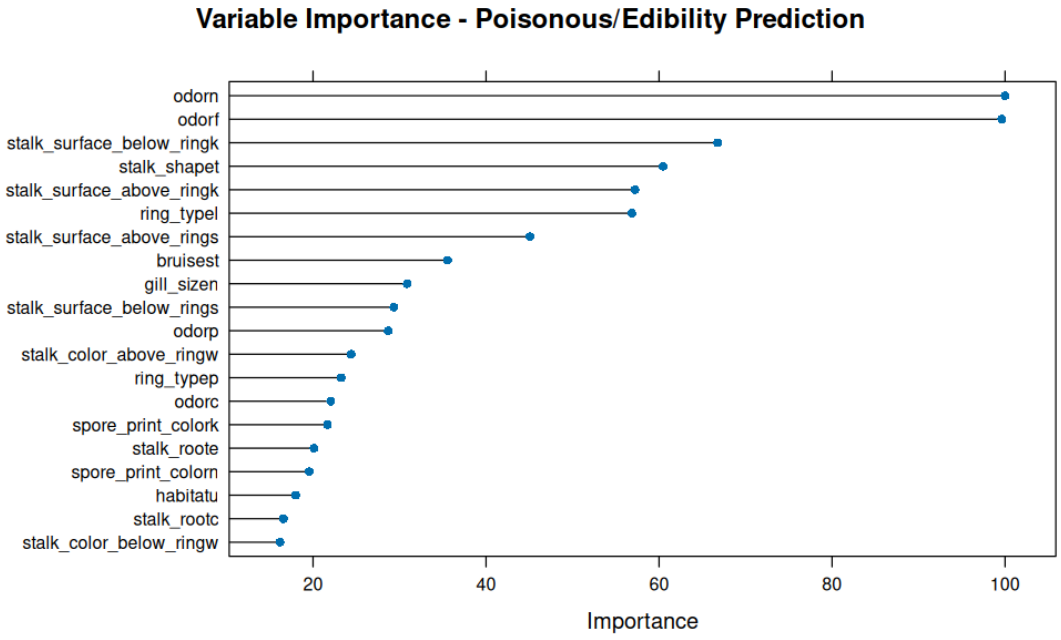

Figure 1: Variable importance for the classification problem to predict poisonousness by random forest predicted from full feature space with 2 predictors sampled for splitting at each node from 5-fold cross validation.

Second, for the odor classification task, we found that edibility—specifically whether a mushroom is poisonous—was the most influential predictor of odor. This suggests that odor is often biologically tied to toxicity, and that poisonous mushrooms tend to emit distinctive smells. Hypothetically, if someone were to lose their sense of smell (e.g., due to illness), they might unknowingly consume a poisonous mushroom—only later recalling the odor based on its other characteristics. Beyond edibility, several morphological features also proved informative for odor prediction, particularly those related to the stalk's structure (i.e., stalk shape and root), as well as gill size. These are observable and tactile traits, implying that even in the absence of direct

olfactory input, one might infer odor through careful visual and physical inspection of a mushroom's morphology.

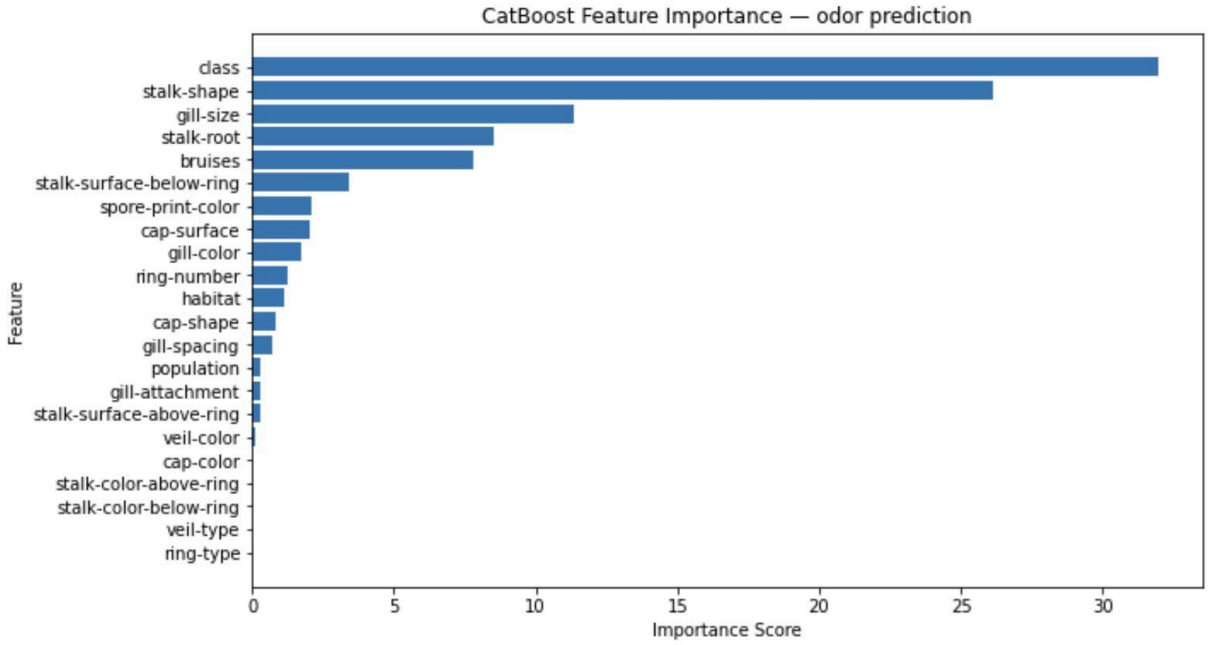

Figure 2: Variable importance for the classification problem to predict odor by CatBoost predicted from full feature space.

Methods

To manage high-dimensional data from categorical features transformed through one-hot encoding, we applied PCA as our dimensionality reduction technique. Starting with 22 original features, dummy encoding significantly expanded the feature space, increasing the risk of overfitting and computation time. PCA helped by reducing this expanded space to 11 principal components that retained most of the variance. This method was chosen because it is unsupervised, making it suitable for a classification task where we didn't want to bias the feature transformation with label information. PCA also handles sparse data well and improves model efficiency and generalization by filtering out noise and redundancy.

To train and tune models to predict poisonousness, we explored the following hyperparameter choices across six models:

- Decision Tree: We tuned cost complexity, tree depth, and minimum number of observations that must exist in a terminal leaf node (min_n). Pre principal component analysis, the most performing hyperparameter set was cost_complexity=0.01, depth=3, min_n =5, yielding an accuracy of 0.9972 and area under receiver operating curve of 0.9964. Post-PCA, we switched to cost_complexity=0.001, depth=7, and min_n=5, optimizing for higher cross-validated accuracy and area under receiver operating curve (both ~0.998–0.999).

- Random Forest: We tuned mtry, the number of features sampled at each split. Before PCA, mtry=2 provided the best performance. After principal component analysis, we increased it to mtry=3 to align with the reduced feature space. In both cases, the model reached near-perfect metrics (area under receiver operating curve and accuracy ≈ 1.0).
- CatBoost: We evaluated various tree depths. Since accuracies strictly increase towards perfect accuracy (1.0000) across increasing depths, we opted for the smallest depth that achieves perfect accuracy. Such values are depth = 2 pre-PCA and depth = 6 post-PCA. Deeper logic trees are required for PCA-reduced classification to achieve the same accuracy.
- ANN: We tuned the number of units in the first hidden layer. Before PCA, 64 units provided the best balance between performance and training time. After PCA, only 16 units were needed to achieve the same accuracy, suggesting that PCA improved training efficiency. This is likely due to the continuous and compact nature of PCA-transformed features, which can make ANN learning more efficient.
- K-Nearest Neighbors (KNN): The hyperparameter is k, the number of nearest neighbors when making a prediction. Initially, with categorical features, we used Hamming distance and chose k=5 (mean cross-validation accuracy = 1). After PCA, where features became continuous, we used Euclidean distance and selected k = 3 (mean cross-validation accuracy = 1). In both cases, the selected k values not only yielded the highest accuracy but were also the largest among all values with equivalent top accuracy. We deliberately chose the larger k (under same accuracy) to encourage better generalization and reduce the risk of overfitting.
- Naive Bayes Classifier: Before applying PCA, we used the categorical Naive Bayes classifier, as the features were all categorical. We tuned the Laplace smoothing parameter α and selected $\alpha = 0.001$, which yielded the highest mean cross-validation accuracy of 0.9973. After PCA, the features were transformed into continuous variables, prompting a shift to Gaussian Naive Bayes. In this context, we instead tuned the var_smoothing parameter, which plays a similar role by preventing numerical instability due to near-zero variance estimates. The value with highest mean cross-validation accuracy (0.962) was 1e-2, which balanced model stability and accuracy.

To train and tune models to predict odor, we explored the following hyperparameter choices across six models:

- Decision Tree: We tuned cost complexity, tree depth, and minimum number of observations that must exist in a terminal leaf node (min_n). Before principal component analysis, the best-performing hyperparameter set was cost_complexity = 0.005, depth = 7, and min_n = 5, which yielded high cross-validated accuracy (0.9972) and area under the receiver operating curve (0.9964). After principal component analysis, we retained the same hyperparameters (cost_complexity = 0.005, depth = 7, min_n = 5) as they continued to offer strong performance (accuracy ≈ 0.9151, area under receiver operating curve ≈ 0.9718) on the transformed features.
- Random Forest: We tuned mtry, the number of features sampled at each split. Before principal component analysis, we evaluated a wide range of values and found that mtry = 8 provided the lowest cross entropy and balanced performance across all metrics, including accuracy (0.8996), area under receiver operating curve (0.9825), and mean

F1-score (0.7943). After principal component analysis, with the reduced feature space, we again selected mtry = 8 as it minimized cross-entropy loss and maintained strong generalization (accuracy ≈ 0.8762, area under receiver operating curve ≈ 0.9792).

- CatBoost: We tuned the depth of the decision trees. PCA reduced accuracy for all hyperparameter values. Then, accuracies fluctuated irregularly without PCA and strictly decreased with PCA. Before PCA, depth = 4 gave the highest mean validation accuracy (0.9214). Post-principal component analysis, depth = 2 yielded the best accuracy (0.9100).
- Artificial Neural Network (ANN): We tuned the number of units in the hidden layer. Accuracies fluctuate irregularly across hyperparameter values, and PCA reduces accuracies for all hyperparameter values. Before PCA, a hidden layer with 64 units gave the highest validation accuracy (0.9252). After principal component analysis, we found that a smaller network with 16 units achieved a comparable validation accuracy (0.9240).
- K-Nearest Neighbors (KNN): Same as predicting poisonousness, we tuned k, the number of nearest neighbors. Before PCA, we used Hamming distance due to the categorical nature of features and selected k = 41 as it provided the highest accuracy (mean cross-validation accuracy = 0.9220). After PCA, since the features became continuous, we switched to Euclidean distance and selected k = 46, the value with highest mean cross-validation accuracy (0.9305).
- Naive Bayes Classifier: Same as predicting poisonousness, before PCA, we used categorical Naive Bayes and tuned the Laplace smoothing parameter alpha, α and selected $\alpha = 0.001$, which led to the best mean cross-validation accuracy (0.9135). After PCA, we transitioned to Gaussian Naive Bayes to handle continuous inputs and tuned var_smoothing over a logarithmic scale. While 1e-4 and 1e-3 both reached the same highest mean cross-validation accuracy (0.9069), we chose the 1e-4 as the optimal value, as smaller smoothing values (closer to zero) helped preserve the structure of the class-conditional distributions while still ensuring numerical stability. Larger values, in contrast, possess higher risk for over-smoothing.

As analyzed above, hyperparameter tuning across models is data-driven and performance-focused, with cross-validation guiding our decisions to optimize accuracy, area under the receiver operating curve, and computational efficiency.