Task 1: Machine Learning on Tabular Mushrooms

The aim of this first task is to use machine learning techniques on a tabular dataset to determine if a mushroom is edible or poisonous. The dataset contains various attributes that describe the physical characteristics of mushrooms, such as shape, color, odor, and habitat. To ensure accurate classification, we have utilized various data preprocessing techniques, model selection, and evaluation metrics.

During our exploration of the dataset, we noticed missing values represented by "?" in the "stalk root" attribute. Instead of removing instances with missing values, which may result in data loss, we decided to treat the missing values as a separate category called "unknown." This allowed us to incorporate the unknown stalk root information into our classification models.

To convert categorical data into a suitable format for machine learning algorithms, we opted for one-hot encoding. This technique creates binary features for each category within the original feature, with 1 indicating the presence of the category and 0 indicating its absence. Although we initially considered using LabelEncoder, which assigns a numerical label to each category, one-hot encoding was a better choice because it avoids implying any ordinal relationship between categories that may not exist.

Once we preprocessed the data, we split it into a training set and a test set using the train_test_split function from the sklearn library. This function randomly divides the dataset into training and test subsets, ensuring that both sets have a similar distribution of instances. We allocated 80% of the data for training and 20% for testing.

We experimented with three different machine learning models: K-Nearest Neighbors (KNN), Logistic Regression, and Neural Network. Each model has unique characteristics and assumptions that can affect their performance on the given dataset.

K-Nearest Neighbors (KNN) is a powerful yet straightforward algorithm that classifies instances based on their proximity to the K nearest instances in the training set. The primary concern when using KNN is selecting an appropriate value for K, which affects the model's bias-variance trade-off. We chose K=5 as a reasonable starting point for our analysis.

Logistic Regression is a linear model for binary classification that estimates the probability of an instance belonging to a specific class. It assumes that the relationship between the features and the log odds of the target variable can be modeled using a linear function. Logistic Regression inherently includes L2 regularization, which helps to prevent overfitting by penalizing large weights in the model.

Neural Networks are a class of models that can learn complex non-linear relationships between features and the target variable. They consist of interconnected layers of nodes, with each layer transforming the input data into a higher level of abstraction. The choice of the architecture, such as the number of layers and nodes, can significantly impact the model's performance. We used a relatively simple neural network architecture with two hidden layers, each containing 128 nodes, suitable for our dataset.

After training each model on the preprocessed data, we evaluated their performance on the test set using accuracy as the primary evaluation metric. While accuracy is a widely used metric, it may not be suitable for all classification problems, particularly when dealing with imbalanced datasets. Other evaluation metrics, such as precision, recall, F1-score, or the area under the Receiver Operating Characteristic (ROC) curve, may provide a more comprehensive understanding of the models' performance.

To further enhance the classification pipeline, we can consider the following steps and techniques:

1. Feature selection or dimensionality reduction: This involves using techniques such as Principal Component Analysis (PCA) or Recursive Feature Elimination (RFE) to reduce the number of features while still retaining the most relevant information.
2. Hyperparameter tuning: We can systematically search for the best hyperparameters by using techniques such as grid search or libraries like Optuna. This can lead to optimized model performance.
3. Cross-validation: By employing k-fold cross-validation, we can obtain a more accurate estimate of model performance and reduce the likelihood of overfitting.
4. Ensemble methods: Combining multiple models, such as bagging, boosting, or stacking, can improve overall performance and increase the robustness of predictions.
5. Model interpretability: We can use techniques such as SHapley Additive exPlanations (SHAP) or Local Interpretable Model-agnostic Explanations (LIME) to help explain the predictions of the models and identify the most important features.

In conclusion, our analysis demonstrates a comprehensive approach to machine learning on tabular data for the task of classifying mushrooms as edible or poisonous. By incorporating additional steps and considerations like feature selection or dimensionality reduction, hyperparameter tuning, cross-validation, ensemble methods, and model interpretability, we can enhance the classification pipeline and achieve a more robust and accurate solution. This approach can be applied to other classification tasks, providing valuable insights and predictions for various domains.